Production Planning and Processing

July 2006

Raytheon

Overview of Lesson

- Introduction
- Production Planning and Processing (Context)
- Production Requests (Including Production Rules)
- Logging in to System Hosts
- Launching the Production Request Editor
- Creating/Updating/Deleting a Production Request
- Reviewing/Deleting Data Processing Requests
- Launching Planning Workbench-Related GUIs

Overview of Lesson (Cont.)

- Creating a New Production Plan (Including Defining a Production Strategy and Reviewing a Plan Timeline)
- Cleaning the PDPS Database and Science Processing Disks
- Troubleshooting Production Planning Problems
- Launching the AutoSys GUI Control Panel
- Configuring AutoSys Screens/Displays
- Reviewing Hardware Status
- Monitoring/Controlling Job Processing
- Tuning System Parameters

Overview of Lesson (Cont.)

- Troubleshooting Processing Problems
- Launching the Quality Assurance (QA) Monitor
- Performing Science Product Quality Assurance
- Regenerating Granules in Response to a Loss of Files from the Archive

Overview of Lesson (Cont.)

- Practical Exercise
  - Logging in to System Hosts
  - Launching the Production Request Editor
  - Creating a New Production Request Using the Production Request Editor GUI
  - Creating a New Production Request Using the Production Request Generator (Command-Line Interface)
  - Editing/Modifying a Production Request
  - Deleting a Production Request
  - Reviewing Data Processing Requests
  - Deleting a Data Processing Request
  - Launching the Production Strategies GUI
  - Launching Production Workbench-Related GUIs

Overview of Lesson (Cont.)

- Practical Exercise (Cont.)
  - Defining a Production Strategy
  - Creating a New Production Plan
  - Reviewing a Plan Timeline
  - Cleaning the PDPS Database and Science Processing Disks
  - Troubleshooting Production Planning Problems
  - Launching the AutoSys GUI Control Panel
  - Configuring AutoSys Runtime Options
  - Reviewing Hardware Status (Including Changing Hardware Status Views)
  - Monitoring/Controlling Job Processing
  - Responding to Alarms (Including Selecting Alarms to Be Displayed)

Overview of Lesson (Cont.)

- Practical Exercise (Cont.)
  - Specifying Job Selection Criteria and Reviewing Job Activities
  - Determining the Ownership of an AutoSys Job
  - Modifying Job Status (Including Sending an Event to a Job) Using AutoSys
  - Reviewing Activity Reports and Job Dependency Reports
  - Defining and Running Monitors/Browsers
  - Troubleshooting Processing Problems
  - Launching the QA Monitor GUI
  - Updating QA Metadata
  - Regenerating Granules in Response to a Loss of Files from the Archive

Objectives

- OVERALL:
  - Develop proficiency in the procedures that apply to production planning and production processing operations
- SPECIFIC:
  - Describe the general functions and processes included in the Planning and Data Processing Subsystems (in the context of system operations)
  - Perform the steps involved in...
    - logging in to system hosts
    - launching the production request editor
    - creating a new production request using the Production Request Editor GUI
    - creating a new production request using the Production Request Generator (command-line interface)

Objectives (Cont.)

- SPECIFIC (Cont.):
  - Perform the steps involved in...
    - modifying a production request
    - deleting a production request
    - reviewing data processing requests
    - deleting a data processing request
    - launching the production strategies GUIs
    - launching planning workbench-related GUIs
    - defining a production strategy
    - creating a new production plan
    - reviewing a production plan timeline
    - cleaning the PDPS database and science processing disks
    - troubleshooting production planning problems

Objectives (Cont.)

- **SPECIFIC (Cont.):**
  - **Perform the steps involved in...**
    - **launching the AutoSys GUI Control Panel**
    - **configuring AutoSys runtime options**
    - **reviewing hardware status (including changing hardware status views) using AutoSys**
    - **monitoring/controlling job processing**
    - **troubleshooting processing problems**
    - **launching the QA Monitor GUI**
    - **updating QA metadata**
    - **regenerating granules in response to a loss of files from the archive**

Objectives (Cont.)

- **STANDARDS:**
  - **Lesson content (procedures in the lesson)**
  - ***Mission Operation Procedures for the EMD Project* (611-EMD-001)**

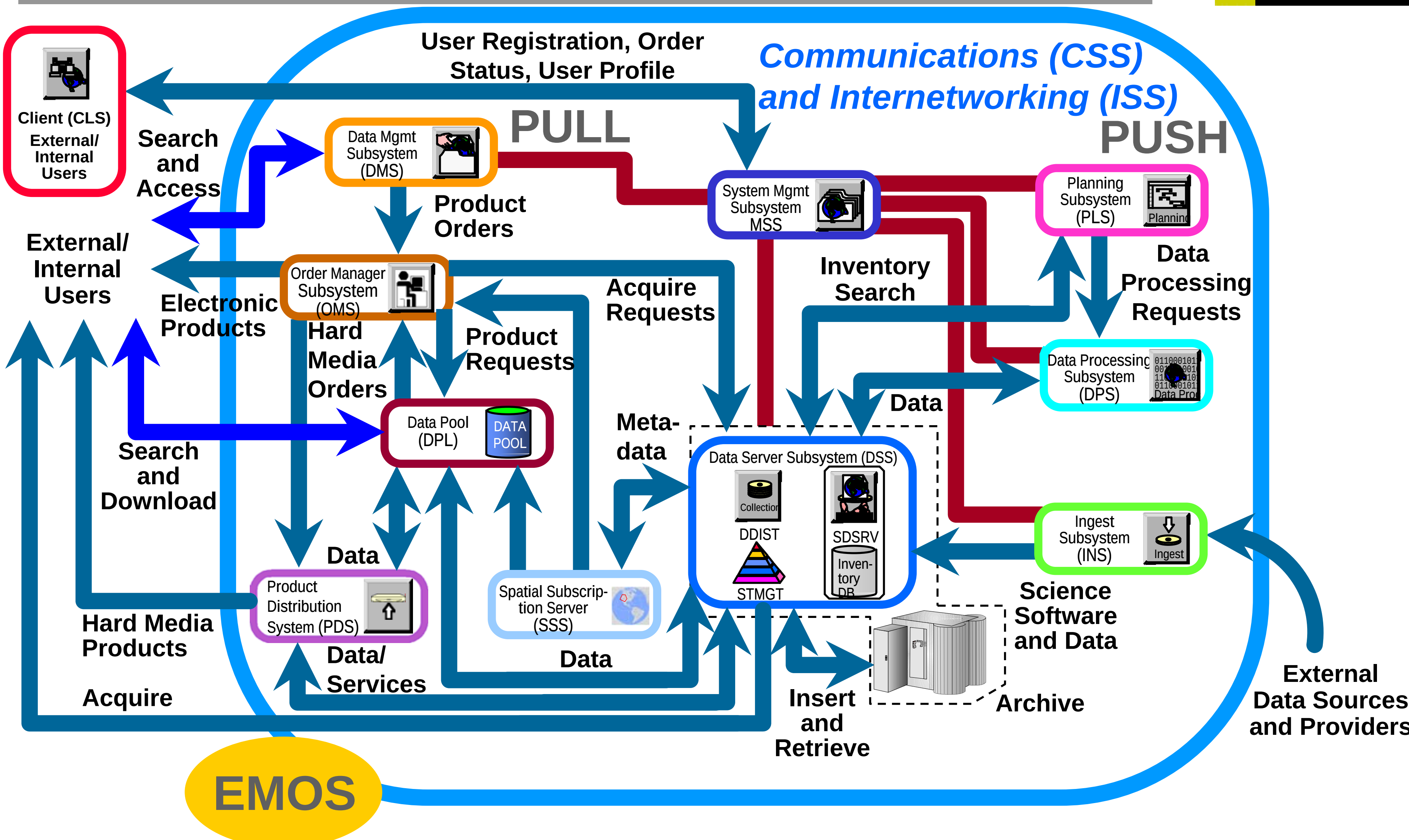
Production Planning and Processing

- **System Context**
  - **Production planning and processing processes are accomplished at the Distributed Active Archive Centers (DAACs)**
  - **People involved in production planning and processing activities are...**
    - **Production Planners**
    - **Production Monitors**

Production Planning and Processing (Cont.)

- System Context (Cont.)
  – Production Planner
    - performs planning functions; especially, using the Planning Subsystem (PLS) to create Data Processing Requests and specify which requests are to be processed as part of a particular Production Plan
  – Production Monitor
    - keeps track of operations in the Data Processing Subsystem, especially the execution of science data processing jobs (creation of data products)

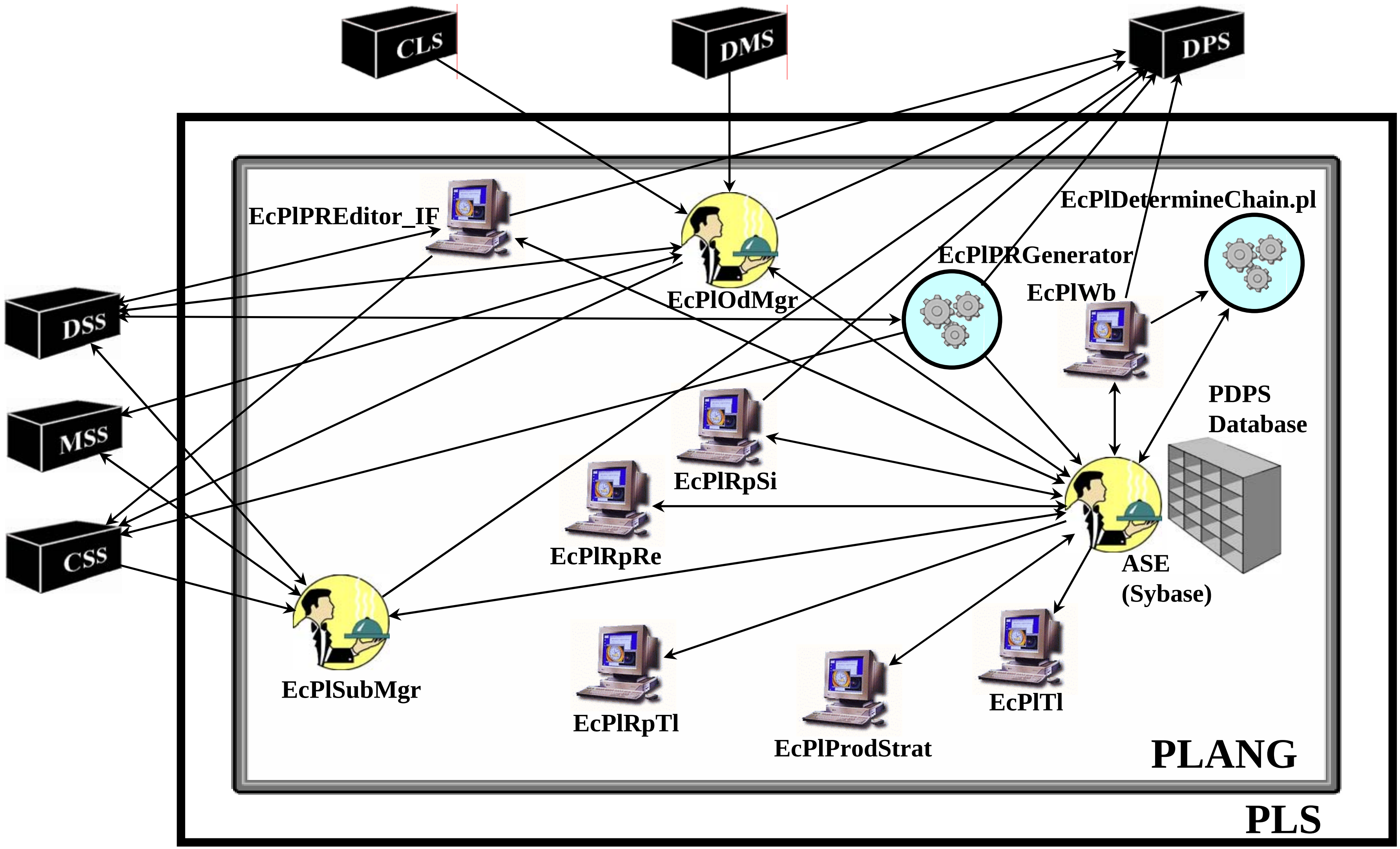
System Context Diagram



Production Planning and Processing (Cont.)

- **Planning Subsystem**
  - provides a mechanism for accomplishing the following general functions:
    - Defining DAAC production resources
    - Scheduling production resources for non-production-related activities
    - Defining data processing jobs to be performed at the DAAC
    - Generating efficient plans for scheduling defined data processing jobs
    - Coordinating production with the Data Server Subsystem and Data Processing Subsystem to achieve a highly automated production system

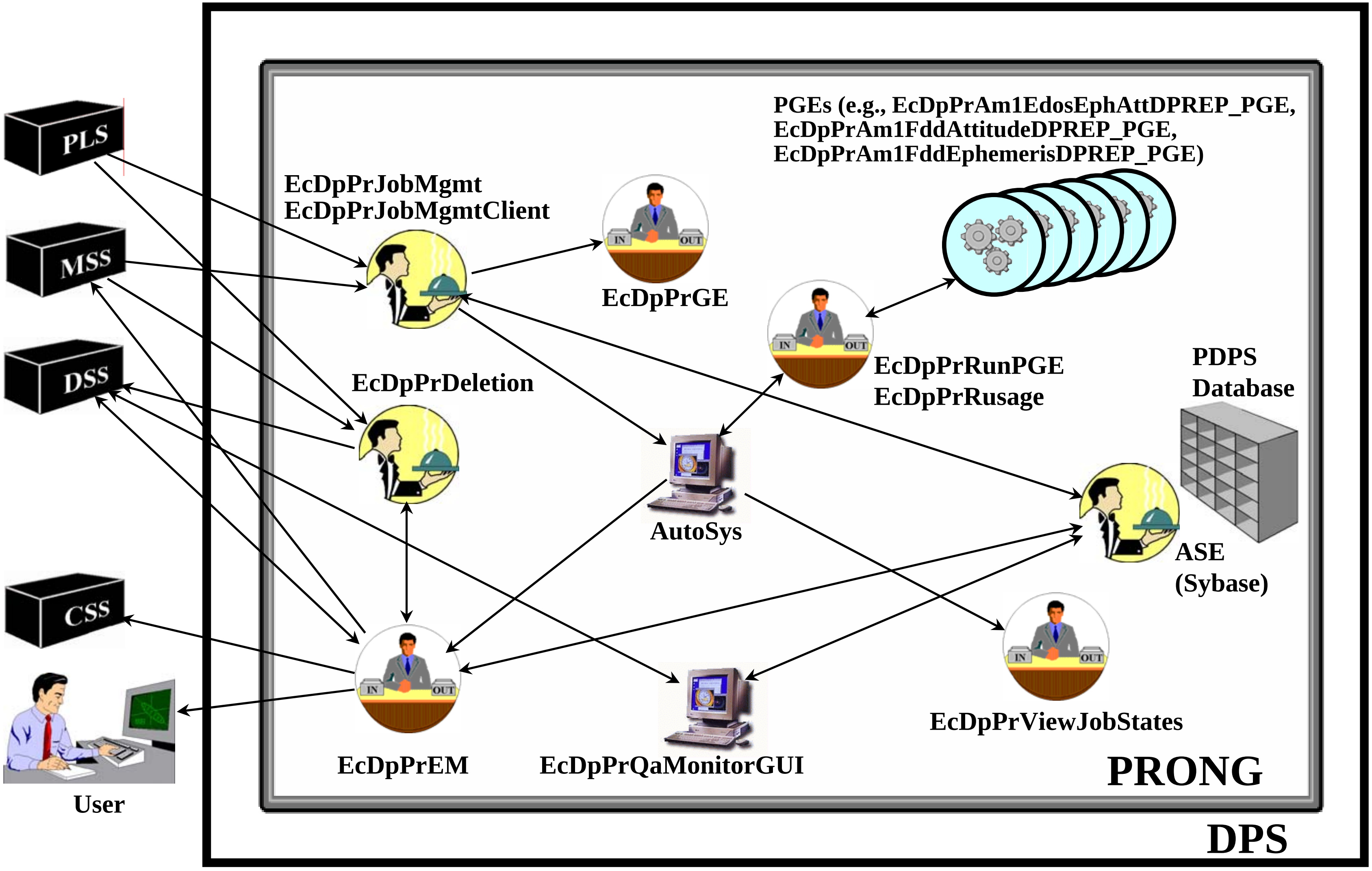
Planning Architecture



Production Planning and Processing (Cont.)

- **Data Processing Subsystem**
  - provides a mechanism for accomplishing the following general functions:
    - Managing the allocation of data processing jobs to the site's data processing resources
    - Managing, queuing, and executing data processing jobs to produce data products
    - Supporting preliminary processing of ancillary data granules

Data Processing Architecture (PRONG)



Production Planning and Processing (Cont.)

- PLANG is the Planning Subsystem computer software configuration item (CSCI)
  - Resource Planning Workbench
    - Resource Editor (EcPlRpRe)
    - Resource Scheduler (EcPlRpSi)
    - Resource Reservation Planning Master Timeline GUI (EcPlRpTl)
  - Production Request Editor (EcPlPREditor)
  - Production Planning Workbench
    - Planning Workbench GUI (EcPlWb)
    - Production Strategies GUI (EcPlProdStrat)
    - Planning Master Timeline GUI (EcPlTl)

Production Planning and Processing (Cont.)

- PLANG (Cont.)
  - On-Demand Manager (EcPlOdMgr)
  - On-Demand Manager Client (EcPlOdMgrClient)
  - Subscription Manager (EcPlSubMgr)
  - Sybase Adaptive Server Enterprise (ASE) Server
  - Message Handler (EcPlMsh)
  - System Name Server (EcPlSns)
  - Resource Model (EcPlRpRm, EcPlRm)

Production Planning Concepts (Cont.)

- **PLANG (Cont.)**
  - **Start-up and shutdown scripts used by planning personnel (/usr/ecs/*MODE*/CUSTOM/utilities directory on the Planning/Management Workstation)**
    - **EcPlSomeStart**
    - **EcPlAllStart**
    - **EcPlPRE_IFStart**
    - **EcPlPRE_ReadOnlyStart**
    - **EcPlPRGeneratorStart**
    - **EcPlProdStratStart**
    - **EcPlRpAllStart**
    - **EcPlRpReStart**
    - **EcPlRpSiStart**
    - **EcPlSubsEditStart**

Production Planning Concepts (Cont.)

- **PLANG (Cont.)**
  - **Start-up and shutdown scripts used by planning personnel (/usr/ecs/*MODE*/CUSTOM/utilities directory on the Planning/Management Workstation) (Cont.)**
    - **EcPlTlStart**
    - **EcPlWbStart**
    - **EcPlSlay**
    - **EcPlSlayAll**
    - **EcPlRpSlayAll**

Production Planning Concepts (Cont.)

- **PLANG (Cont.)**
  - **Start-up scripts in the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Queuing Server:**
    - **EcPlOdMgrClientStart**
    - **EcPlOdMgrStart**
    - **EcPlPlanningAppStart**
    - **EcPlStart**
    - **EcPlSubMgrStart**

Production Planning Concepts (Cont.)

- **PLANG (Cont.)**
  - **Start-up scripts called by other applications (not normally invoked directly by planning personnel)**
    - **EcPlMshStart**
    - **EcPlRmStart**
    - **EcPlRpRmStart**
    - **EcPlSnsStart**
    - **EcPlStart**
    - **SweeperStart**

Production Planning Concepts (Cont.)

- **PLANG (Cont.)**
  - **Other scripts**
    - **EcLgLogCtrlStart**
    - **EcPlCdsPingServers**
    - **EcCsPerfLogProcessor.pl**
    - **EcPlDbClean**
    - **EcPlDbCleanArchive**
    - **EcPlDbBuild**
    - **EcPlDbDrop**
    - **EcPlDbDump**
    - **EcPlDbMigrate**
    - **EcPlDbPatch**
    - **EcPlDetermineChain.pl**
    - **EcPlRpFetchBaseline**

Production Planning Concepts (Cont.)

- **PLANG (Cont.)**
  - **Other scripts (Cont.)**
    - **EcPlDbReset**
    - **EcPlDbList**
    - **EcPlDbSave**
    - **fos_services**

Production Planning and Processing (Cont.)

- Data Processing Subsystem is composed of three computer software configuration items (CSCIs):
  - PRONG
    - Provides the services required to manage and monitor the Science Data Processing environment, which executes Science Software items (PGEs) and produces data products
  - Algorithm Integration & Test Tools (AITTL)
    - Set of tools used for test and integration of new science software, new versions of science software, and user methods into the Science Data Processing operational environment
  - Science Data Processing (SDP) Toolkit
    - Provides a set of software libraries which are used to integrate Science Software into the system environment

Production Planning and Processing (Cont.)

- **PRONG CSCI**
  - Job Management (EcDpPrJobMgmt)
  - Ground Event process (EcDpPrGE)
  - Job Management Client (EcDpPrJobMgmtClient)
  - AutoSys
    - Event Processor (AutoSys daemon)
    - Event Server
    - AutoSys GUIs
    - AutoSys Job Management Web Interface
  - Execution Management (EcDpPrEM)
  - PGE Execution Manager (EcDpPrRunPGE)
  - Resource Usage (EcDpPrRusage)

Production Planning and Processing (Cont.)

- **PRONG CSCI (Cont.)**
  - **View Job States (EcDpPrViewJobStates)**
  - **Terra Data Preprocessing (DPREP)**
    - **EcDpPrAm1EdosEphAttDPREP_PGE**
    - **EcDpPrAm1FddAttitudeDPREP_PGE**
    - **EcDpPrAm1FddEphemerisDPREP_PGE**
    - **EcDpPrDumpAttitudeDPREP**
    - **EcDpPrDumpEphemerisDPREP**

Production Planning and Processing (Cont.)

- PRONG CSCI (Cont.)
  - Aqua DPREP
    - EcDpPrPm1FddEphemerisDPREP_PGE
    - EcDpPrPm1AttitudeDPREP_PGE
  - Aura DPREP
    - EcDpPrAuraEphemerisDPREP_PGE
    - EcDpPrAuraAttitudeDPREP_PGE
  - Deletion Server (EcDpPrDeletion)
  - Deletion Client (EcDpPrDeletionClient)
  - Sybase ASE Server
  - Quality Assurance Monitor (EcDpPrQaMonitorGUI)
  - EcDpPrLoadTable.pl
  - EcDpPrAutocons

Production Planning and Processing (Cont.)

- **PRONG CSCI (Cont.)**
  - EcDpPrEMGetAncHeaders
  - EcDpPrPREPQCConverterPGE
  - EcDpPrReadPREPQCData
  - EcDpPrWritePREPQCDataToHDFEOS
  - EcDpPrSMFCopy

Production Planning Concepts (Cont.)

- PRONG (Cont.)
  – Start-up scripts used by production personnel (/usr/ecs/*MODE*/CUSTOM/utilities directory on the Queuing Server)
    - EcDpPrAutosysStart
    - EcDpPrDeletionClientStart
    - EcDpPrGarbageCollectorStart
  – Start-up scripts called by other applications (not normally invoked directly by production personnel)
    - EcDpPrDeletionStart
    - EcDpPrDisplayJobStates
    - EcDpPrJobMgmtClientStart
    - EcDpPrJobMgmtStart
    - EcDpPrStart
    - EcDpProcessingAppStart

Production Planning Concepts (Cont.)

- **PRONG (Cont.)**
  - **Start-up script used by production personnel (/usr/ecs/*MODE*/CUSTOM/utilities directory on the Planning/Management Workstation)**
    - **EcDpPrQaMonitorGUIStart**
  - **Other scripts available on the Queuing Server:**
    - **EcDpBusySystemClean.pl**
    - **EcDpPrRestartFailedJobs**
    - **EcDpPrCleanMaintMachineFiles.pl**
    - **EcDpPrRmFilesWOGranules.pl**

Production Requests

- Production planning at the Distributed Active Archive Centers (DAACs)
  - process by which the Production Planner notifies the Planning Subsystem (PLS) of the science processing jobs that are to be processed and what data to process
- Science Software (SS)
  - does the actual data processing
  - is developed at Science Computing Facilities (SCFs)
  - is embodied in Product Generation Executives (PGEs) when the software is integrated into the system production processing environment

Production Requests (Cont.)

- PGEs
  – science software code (e.g., executable programs or shell scripts) that contain the instructions for processing data to create the desired products
- Production Request (PR)
  – Production Planner defines science data processing jobs in terms of PRs
  – A PR is an order for data to be produced by the data processing system
  – A single PR may specify…
    - several jobs that are to be run over a period of time
    - a single job producing a single set of data

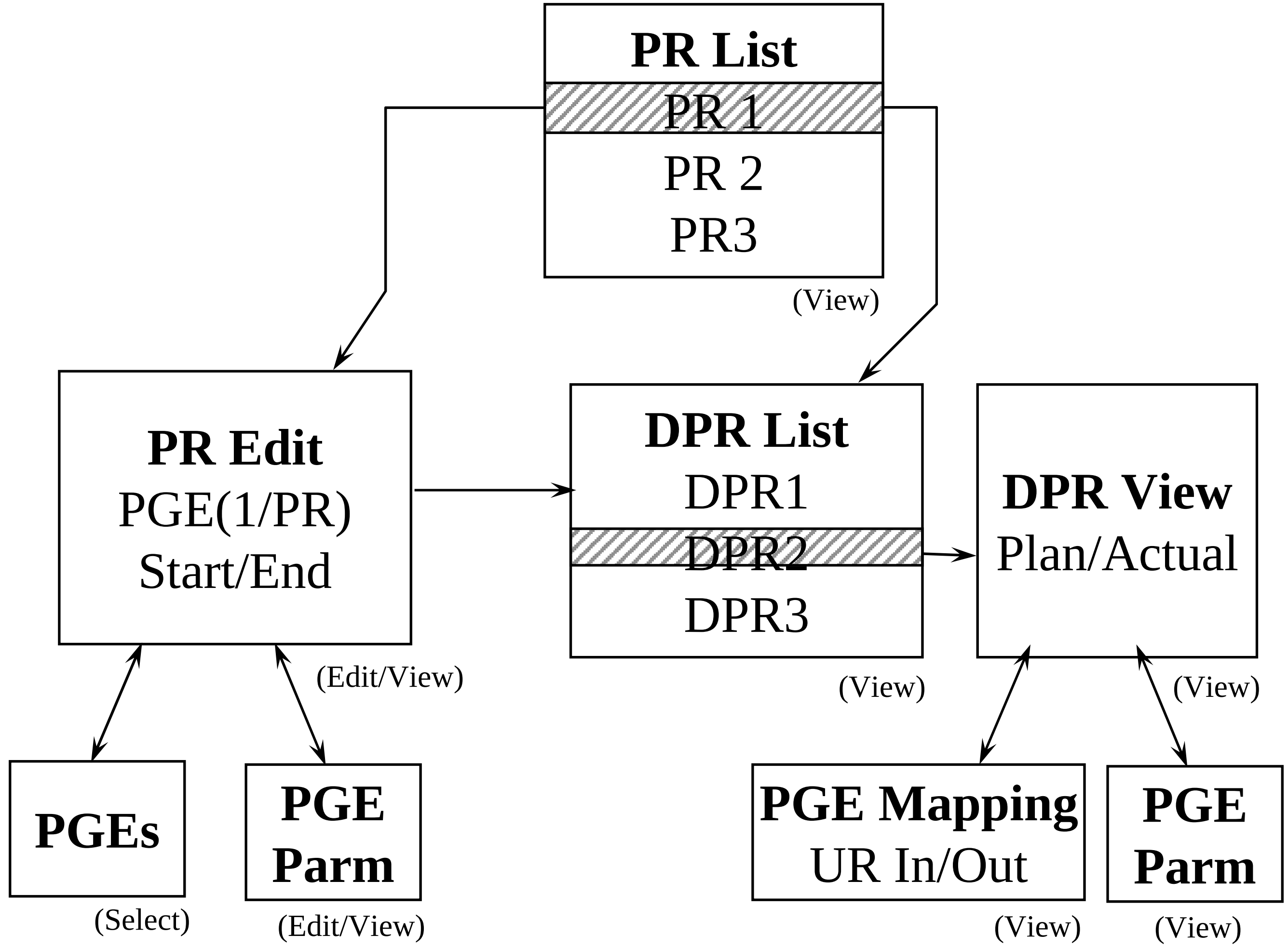
Production Requests (Cont.)

- **Production Request (Cont.)**
  - **may apply to the processing of new data (standard PRs or standing orders)**
  - **may apply to the reprocessing of existing data (reprocessing PRs)**
  - **each PR identifies a specific PGE for generating a particular type of product**
    - **some PGEs are dependent on others; i.e., some PGEs require input data that are the output of other PGEs**
  - **planning software will recognize and reject a PR when the PR specifies a PGE that requires data from another PGE that has not yet been specified in a PR**

Production Requests (Cont.)

- **Planning Subsystem**
  - uses each PR to generate either one or a series of Data Processing Requests (DPRs)
    - each DPR corresponds to one execution of a single PGE
    - each DPR contains the information that is needed by the SDPS processing function
  - checks the availability of the data required for the DPR
  - determines what data will be included in the DPR output
    - so the system can make predictions concerning the future availability of data

Production Requests (Cont.)



Production Requests (Cont.)

- Types of Processing
  - Routine Processing
    - pre-defined software production processing that is periodic and keyed to data arrival
  - Reprocessing
    - using a new, improved PGE to process data that had previously been processed with an older version of the PGE
  - Regeneration
    - a type of reprocessing performed for the purpose of replacing a missing or damaged product
  - On-Demand Processing
    - ad-hoc processing initiated by an end-user (as opposed to the Production Planner)

Production Requests (Cont.)

- **SCF Processing Requests**
  - **Reference:**
    - ***Interface Control Document Between EOSDIS Core System (ECS) and Science Computing Facilities (SCF)* [505-41-33]**
  - **SCFs may request the DAACs to process or reprocess data produced using SCF-provided science software**
  - **SCFs' processing/reprocessing request messages can include optional run-time parameters, in case of changes from the original processing parameters**

Production Requests (Cont.)

- SCF Processing Requests (Cont.)
  - SCF staff send Processing Requests to the DAAC staff using one of the following three protocols:
    - e-mail
    - X11/internet
    - X11/modem
  - SCFs must use the e-mail interface unless the DAAC has authorized their use of the X11 Interfaces
    - Specific protocol used in each instance will have been defined in an Operations Agreement (or lower-level type of document) between SCF and DAAC

Production Requests (Cont.)

- SCF Processing Requests (Cont.)
  - SCF Processing Requests provide DAAC operations staff with the following information:
    - PGE Name
    - PGE Version
    - Optional PGE Profile (number 1-99 defining how the PGE is to be run)
    - Time window for which the product should be regenerated
    - Optional run-time parameters
    - Optional comment
  - DAAC operations personnel acknowledge receipt of an SCF's Processing Request by replying to the SCF in an e-mail message that includes the SCF's original request and the following statement:
    - "This is an acknowledgment of the receipt of the following Processing Request: . . . "

Production Requests (Cont.)

- Production Rules
  - Instructions about how a particular PGE is to be run
  - Specify a wide range of information such as…
    - input and output data types
    - frequency of execution
    - activation conditions
    - error handling instructions
  - PGE profiles
    - permit a PGE to use different sets of production rules
    - allow running a PGE with different input data sets, or activation conditions

Production Requests (Cont.)

- Production Rules (Cont.)
  - Are entered when a PGE undergoes Science Software Integration and Test (SSI&T) at the DAAC
  - Where applicable, default parameter values are entered during SSI&T
  - Some parameters can be overridden when a Production Request is entered
  - Types of conditions that can be specified for each PGE:
    - time period for which the PGE will run
    - PGE inputs
    - PGE outputs
    - runtime parameter values
  - Some (but not all) production rules can work with other production rules

Production Requests (Cont.)

- **Production Rules (Cont.)**
  - **Production rules are often used for the selection of dynamic inputs**
    - **dynamic internal inputs are produced by other PGEs (they are called dynamic internal inputs because they are produced within a DAAC)**
    - **dynamic external inputs are periodically ingested and stored in the Data Server Subsystem (they are termed dynamic external inputs because they are produced outside of the DAAC)**
    - **static inputs are granules that are inserted during the SSI&T process and are retrieved not on the basis of time but by Earth Science Data Type (ESDT) and science group**

Production Requests (Cont.)

- **Production Rules (Cont.)**
  - Methods of defining production rules
    - science metadata is entered in various types of files during the SSI&T process
    - parameter values are entered when a Production Request is created to schedule the PGE
  - During SSI&T at the DAAC information concerning the production rule(s) applicable to the PGE is included in Object Description Language (ODL) files
  - ODL files use ‘parameter equals value’ format

Production Requests (Cont.)

- **Production Rules (Cont.) - Categories of ODL files**
  - – **PGE Science Metadata ODL Files**
  - – **ESDT Science Metadata ODL Files**
  - – **Production Rule-Specific Science Metadata ODL Files**
    - - **Orbit Definition ODL Files**
    - - **Path Map Definition ODL Files**
    - - **Tile Science Metadata ODL Files**

Production Requests (Cont.)

- Production Rules
  - Basic Temporal
    - Temporal (time) range of inputs matches the temporal range of outputs
  - Advanced Temporal
    - Temporal range of inputs is offset from the expected temporal range of inputs and outputs
  - Alternate Inputs
    - PGE is run with different inputs based on the availability or quality of various alternate input data sets
  - Optional Inputs
    - PGE is run with specified optional inputs if available; otherwise, PGE is run without them

Production Requests (Cont.)

- Production Rules (Cont.)
  - Minimum/Maximum Number of Granules
    - Minimum number of input granules needed for full data coverage and maximum number of input granules to search for may be specified
    - Minimum and maximum number of outputs expected from the PGE may be specified
  - Optional DPRs
    - The only DPRs executed are those for which the non-routine key input data actually become available (i.e., are either produced in data processing or can be acquired from the archive)

Production Requests (Cont.)

- Production Rules (Cont.)
  - Metadata Checks
    - DPR is run only if input data's metadata value(s) meet(s) certain criteria
  - Metadata Query
    - Input granule selection is based on metadata value
  - Spatial Query/Spatial Pad
    - Input granule selection is based on the spatial coverage of another input (i.e., the key input)
    - Spatial Pad involves adding area to all sides of the key input's spatial shape

Production Requests (Cont.)

- Production Rules (Cont.)
  - Closest Granule
    - DPR is generated if a required input granule within a particular time range (rather than an exact time) is available; otherwise, no DPR is generated
    - Supersedes the Most Recent Granule Production Rule
  - Orbital Processing
    - Selection of input times is based on orbit information
  - Multiple DPRs for Insertion Time
    - Allows the creation of DPRs for multiple granules with the same insertion time (affects ASTER L1B routine processing only)

Production Requests (Cont.)

- **Production Rules (Cont.)**
  - **Tiling**
    - **Input data is chosen on the basis of Instrument Team-defined tiles (geographic areas)**

Production Requests (Cont.)

- **Basic Temporal Production Rule**
  - Defines the time frame for the PGE along with its input and output data
  - Typically scheduled to run using input data that become available periodically (every hour, every day, etc.)
  - Use input data for a particular period of time
  - Produce output for a specified length of time
  - Data the PGE takes in (its input) and the data it produces (its output) have the same period (or some subset of the same period) as the PGE

Production Requests (Cont.)

- **Basic Temporal Production Rule (Cont.)**
  - **Example One:**
    - **A MODIS PGE processes data for five-minute intervals, producing Level 1B granules**
    - **The PGE requires as input the specific five-minute Level 1A granule that is contemporaneous with (covers the same five-minute time period as) the Level 1B granule to be produced**
    - **Using the Basic Temporal Production Rule, a five-minute Level 1A granule is staged as input to the PGE and a five-minute Level 1B granule is expected as output, both matching the time frame for which the PGE is run**

Production Requests (Cont.)

- **Basic Temporal Production Rule (Cont.)**
  - **Example Two:**
    - **A PGE for a different instrument processes data for 24-hour intervals, producing 24-hour Level 1A granules as output**
    - **As input the PGE takes Level 0 data that is ingested every two hours**
    - **Using the Basic Temporal Production Rule, twelve two-hour Level 0 granules are staged as input to the PGE and a 24-hour Level 1A granule is expected as output, matching the time frame for which the PGE is run**

Production Requests (Cont.)

- **Basic Temporal Production Rule (Cont.)**
  - **Fundamental elements used to define the Basic Temporal Production Rule are…**
    - **period**
    - **boundary**
  - **Period is the length of time for which a PGE processes data or the length of time for which input and output data is collected**
    - **A PGE that is subject to the Basic Temporal Production Rule only and that processes data in two-hour blocks, takes in data that relates to a particular two-hour interval and produces output data for that same two-hour period**
    - **Data that has a period of 15 minutes was collected or produced for a 15-minute time period**

Production Requests (Cont.)

- **Basic Temporal Production Rule (Cont.)**
  - **Boundary is the starting point for the data or PGE**
    - **Depending on the characteristics of the data or PGE, the boundary may be the start of a minute or hour or day or week (etc.)**
    - **If a PGE's boundary is the start of the hour, it processes data that starts every hour and runs on data for the length of its period**
    - **If data comes in every day, PDPS predicts that the data is going to be available at the start of the day and allows scheduling of PGEs that use the data as input accordingly**

Production Requests (Cont.)

- **Basic Temporal Production Rule (Cont.)**
  - **Both the PGE itself and the input data have a boundary and period associated with them**
    - **So PDPS can determine the frequency of processing for a Basic Temporal PGE and the time period for its inputs and outputs**
  - **PDPS uses period and boundary in combination to plan the processing of each PGE, including determining its input requirements and anticipated output (which may be input to other PGEs)**
    - **If a PGE has a period of one hour and a boundary of "start of day," it is scheduled every hour, beginning at midnight**
    - **If an input has a period of 15 minutes and boundary of "start of hour," PDPS predicts it every 15 minutes beginning on the hour**

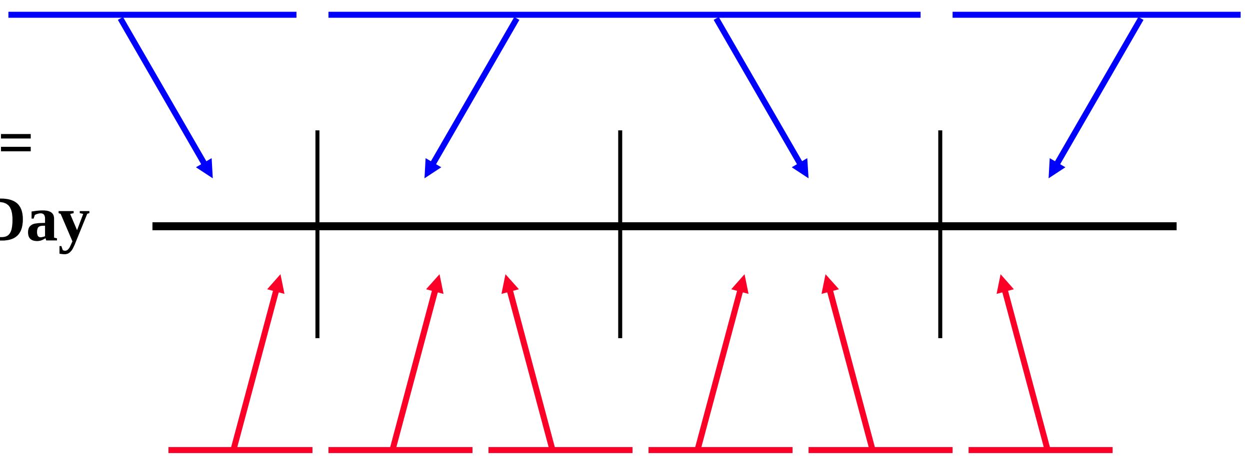
Production Requests (Cont.)

- **Basic Temporal Production Rule (Cont.)**
  - **Additions to the Basic Temporal Production Rule**
    - **boundary offset**
    - **end-of-month anomaly**
  - **Illustration of the Basic Temporal Production Rule**
    - **PGE boundary is "start of day"**
    - **PGE period is one hour**
    - **PGE is scheduled for every hour through the day**
    - **If a Production Request were entered for two full days of processing, a DPR would be created for the PGE to run every hour; i.e., 48 DPRs total**
    - **If a Production Request were created for a four-hour period in the middle of a single day (for example, from 12:00 noon to 4:00 p.m.), then four DPRs would be created**

Example of Basic Temporal Production Rule

Input One: Boundary = Start of Day
Period = 2 hours

PGE
Boundary =
Start of Day
Period =
1 hour



Input Two: Boundary = Start of Day
Period = 1/2 hour

Production Requests (Cont.)

- Advanced Temporal Production Rule
  - Allows for input data to be acquired for a time period other than that of the PGE or its planned inputs/outputs
  - Offset mechanism
    - specifies on an input basis that the data required for processing is some number of seconds earlier or later than the planned time period for the PGE
  - Example One:
    - A PGE requires data from its previous execution for interpolation purposes (e.g., one of its inputs is the output of the very same PGE the last time that it ran)
    - If the PGE processes data for each one-hour interval (producing an hourly product), the Advanced Temporal Production Rule is specified with an offset of minus 3600 seconds (one hour) for the input of the ESDT produced by previous runs

Production Requests (Cont.)

- **Advanced Temporal Production Rule (Cont.)**
  - **Example Two:**
    - **A PGE takes as input two-hour Level 0 data to produce an L1A product**
    - **Because the edges of the Level 0 data can be difficult to process without preceding and succeeding data, the PGE requires three Level 0 granules, one from the time period before it runs, one for the time period it is currently processing and one for the next time period**
    - **The PGE is defined as having three inputs, the first with an Advanced Temporal offset of minus 7200 seconds (two hours), the second with no Advanced Temporal offset and the third with an Advanced Temporal offset of plus 7200 seconds (two hours)**

Production Requests (Cont.)

- **Advanced Temporal Production Rule (Cont.)**
  - **Uses the times specified in the Basic Temporal Production Rule as a reference point for specifying offset(s) to request data from a “period” and/or “boundary” different from that of the DPR or its input**
  - **Offsets ...**
    - **are specified as either negative or positive numbers**
    - **indicate whether the time period of the input data is before or after that of the DPR (a particular run of a PGE)**

Production Requests (Cont.)

- **Advanced Temporal Production Rule (Cont.)**
  - **Begin Period Offset**
    - **an amount of time (in seconds) that is specified with respect to the DPR start time**
    - **negative beginning offset requests data that was collected before the DPR start time**
    - **positive beginning offset requests data with a collection time after the start time of the DPR**
  - **End Period Offset**
    - **an amount of time (in seconds) that is specified with respect to the DPR end time**
    - **negative ending offset requests data that ended collection before the DPR end time was reached**
    - **positive ending offset requests data that ended collection after the end time of the DPR boundaries**
  - **Beginning and ending offsets are not absolute cut-offs for data (overlapping granules are staged)**

Production Requests (Cont.)

- **Advanced Temporal Production Rule (Cont.)**
  - **Illustration of the Advanced Temporal Production Rule**
    - **PGE processes data for every one-hour interval**
    - **Input One comes in at two-hour intervals**
    - **Input Two is produced every 1/2 hour**
    - **Input One: both the Begin Period Offset and End Period Offset are -7200 seconds (minus two hours)**
    - **Every DPR will stage the "previous" Input One (this could be used to get the "previous" or "next" granule of an input)**

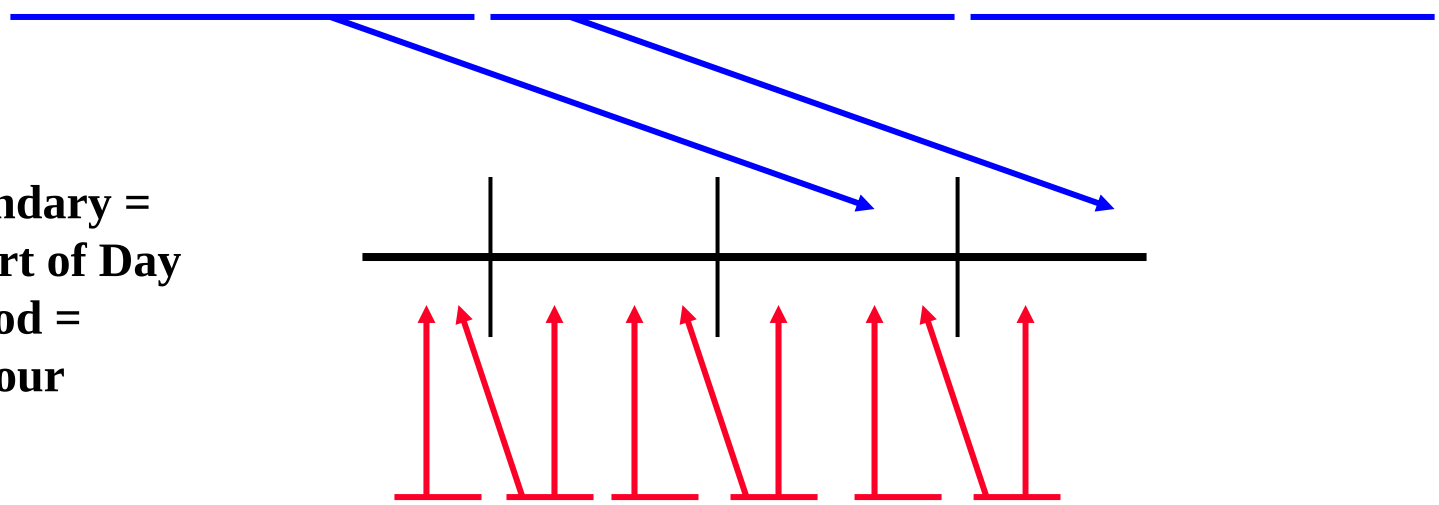
Production Requests (Cont.)

- **Advanced Temporal Production Rule (Cont.)**
  - **Illustration of the Advanced Temporal Production Rule (Cont.)**
    - **Input Two Begin Period Offset for is zero (it will match the Start Time of the DPR)**
    - **Input Two End Period Offset is +1800 seconds (plus 1/2 hour)**
    - **All Input Two granules that fall within the time period of the DPR plus 1/2 hour would be staged**
    - **All Input Two granules within the time period of the DPR are acquired, plus the one from the next 1/2-hour time period, for a total of three granules**
    - **The additional granule acquired by means of the End Period Offset might be used for interpolation purposes at the end point**

Example of the Advanced Temporal Production Rule

Input One: Boundary = Start of Day
Period = 2 hours
Begin Period Offset = -7200 (-2 hours)
End Period Offset = -7200 (-2 hours)

PGE
Boundary =
Start of Day
Period =
1 hour



Input Two: Boundary = Start of Day
Period = 30 minutes (1/2 hour)
Begin Period Offset = 0
End Period Offset = +1800 (+1/2 hour)

Production Requests (Cont.)

- **Alternate Inputs and Optional Inputs Production Rules**
  - **Very similar rules that involve much the same processing in PDPS**
  - **Allow a PGE to select various inputs based on timers and priority lists**
  - **Major difference:**
    - **Alternate Inputs requires that one of alternates on the list be used**
    - **Optional Inputs allows successful execution of the PGE if no optional input on the list is available**

Production Requests (Cont.)

- Alternate Input Production Rule
  - A list of inputs for a PGE is evaluated in priority order
    - PGE is scheduled and executed with the best priority input that could be found
  - A timer can be used to specify how long to wait for a given alternate choice before proceeding with a choice of lesser priority
  - PGE is not executed until one of the alternate choices has been found

Production Requests (Cont.)

- **Alternate Input Production Rule (Cont.)**
  - **Example:**
    - **PGE requires model wind data as an input but is capable of accepting wind data from a Data Assimilation Office (DAO) model, a National Centers for Environmental Prediction (NCEP) model, or climatology**
    - **Each input is considered in priority order**
    - **A timer value indicates how long to wait before trying the next input**
    - **DAO is listed as first choice or "primary" data**
    - **NCEP is the second choice**
    - **Climatology is the last choice**
    - **PGE waits for the DAO timer to expire before running with either NCEP data or climatology**
    - **PGE waits for the NCEP timer to expire before running with the climatology data**

Production Requests (Cont.)

- Optional Input Production Rule
  - There is a list of inputs that are desired but not required for execution of a PGE
  - Inputs are ranked in priority order
  - Timers are set to wait before choosing a lower-priority type of input
  - If none of the inputs on the list becomes available, the PGE starts because the alternatives are classified as "optional”

Production Requests (Cont.)

- **Optional Input Production Rule (Cont.)**
  - **Example:**
    - **It would be preferable to run a particular MODIS PGE with the output of a MISR PGE as input**
    - **However, the MISR output may not be produced every day**
    - **So the MODIS PGE lists the MISR input as optional with a two-hour timer**
    - **On those occasions when no MISR output is produced, the MODIS PGE waits for two hours and then is executed without the MISR input**

Production Requests (Cont.)

- Illustration of the Alternate Inputs and Optional Inputs Production Rules
  - PGE has two inputs that are “required” so they must available for the PGE to be run
  - PGE has one input that is “alternate”
    - Alternate input can be one of three choices
  - After the pair of required inputs has become available, the alternate inputs are evaluated:
    - If the primary alternate is available, it is used as input and the PGE is scheduled for execution
    - One-hour timer on the primary alternate
    - If the primary alternate is unavailable, the PGE waits until the primary alternate becomes available or the one-hour timer expires, whichever occurs first
    - If the second alternate is available after the timer for the primary alternate has expired, the second alternate is used as input

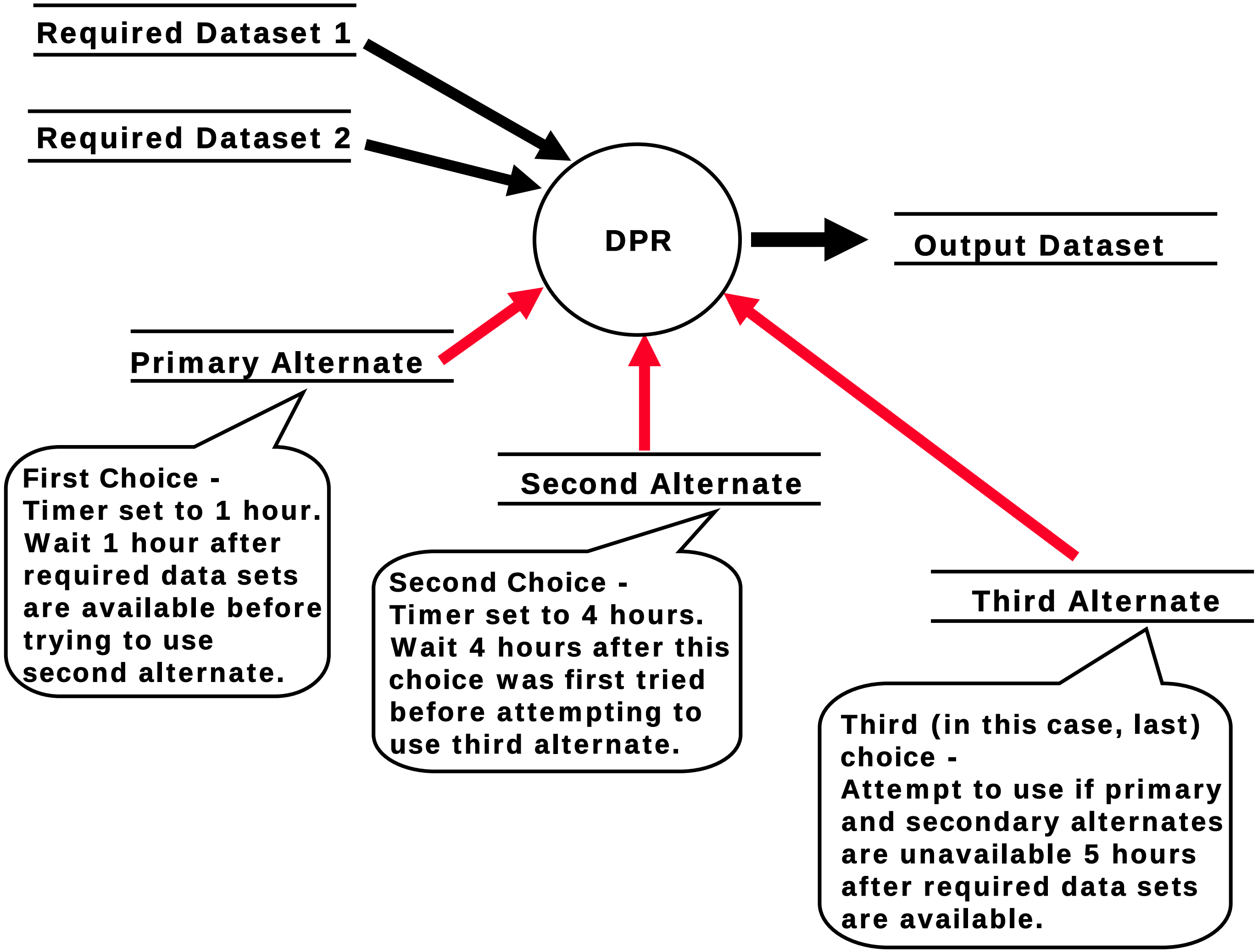
Production Requests (Cont.)

- Illustration of the Alternate Inputs and Optional Inputs Production Rules (Cont.)
  – There is a four-hour timer on the second alternate
  – If the second alternate is unavailable, the PGE waits until either the primary alternate or the secondary alternate becomes available or the four-hour timer expires, whichever occurs first
  – If the third alternate is available after the timer for the second alternate has expired, the third alternate is used
  – There is no timer on the third alternate
  – If the third alternate is not available, the PGE waits until either the primary alternate, the secondary alternate, or the third alternate becomes available, whichever occurs first
  – The PGE does not start processing until one of the alternates becomes available

Production Requests (Cont.)

- **Illustration of the Alternate Inputs and Optional Inputs Production Rules (Cont.)**
  – **If instead of an alternate the third input for the PGE had been defined as an optional input…**
    - **The preceding scenario would have been the same, except that if neither the primary alternate, the second alternate nor the third option was available after the timers had expired, the PGE would not wait**
    - **It would be scheduled for execution without the third input**
    - **It would run with the two required inputs only**

Example of Alternate Inputs Production Rule



Production Requests (Cont.)

- Minimum/Maximum Number of Granules Production Rule
  - Makes it possible to specify a range of possible granules for a given input or output for a PGE
  - Inputs
    - Minimum number of granules the PGE needs for full data coverage
    - Maximum number of granules for the time period
  - Outputs
    - Minimum number of outputs that the PGE is expected to produce
    - Maximum number of outputs that the PGE is expected to produce

Production Requests (Cont.)

- **Minimum/Maximum Number of Granules Production Rule (Cont.)**
  - **Example:**
    - **PGE processes data for every 90-minute interval, has a period of 90 minutes, and takes as input a granule with a period of two hours**
    - **In many instances one granule of the input will satisfy the PGE**
    - **In other instances, because of the way the two-hour and 90-minute periods overlap, the PGE needs two input granules to cover the time period**
    - **Minimum Number of Granules = 1**
    - **Maximum Number of Granules = 2**

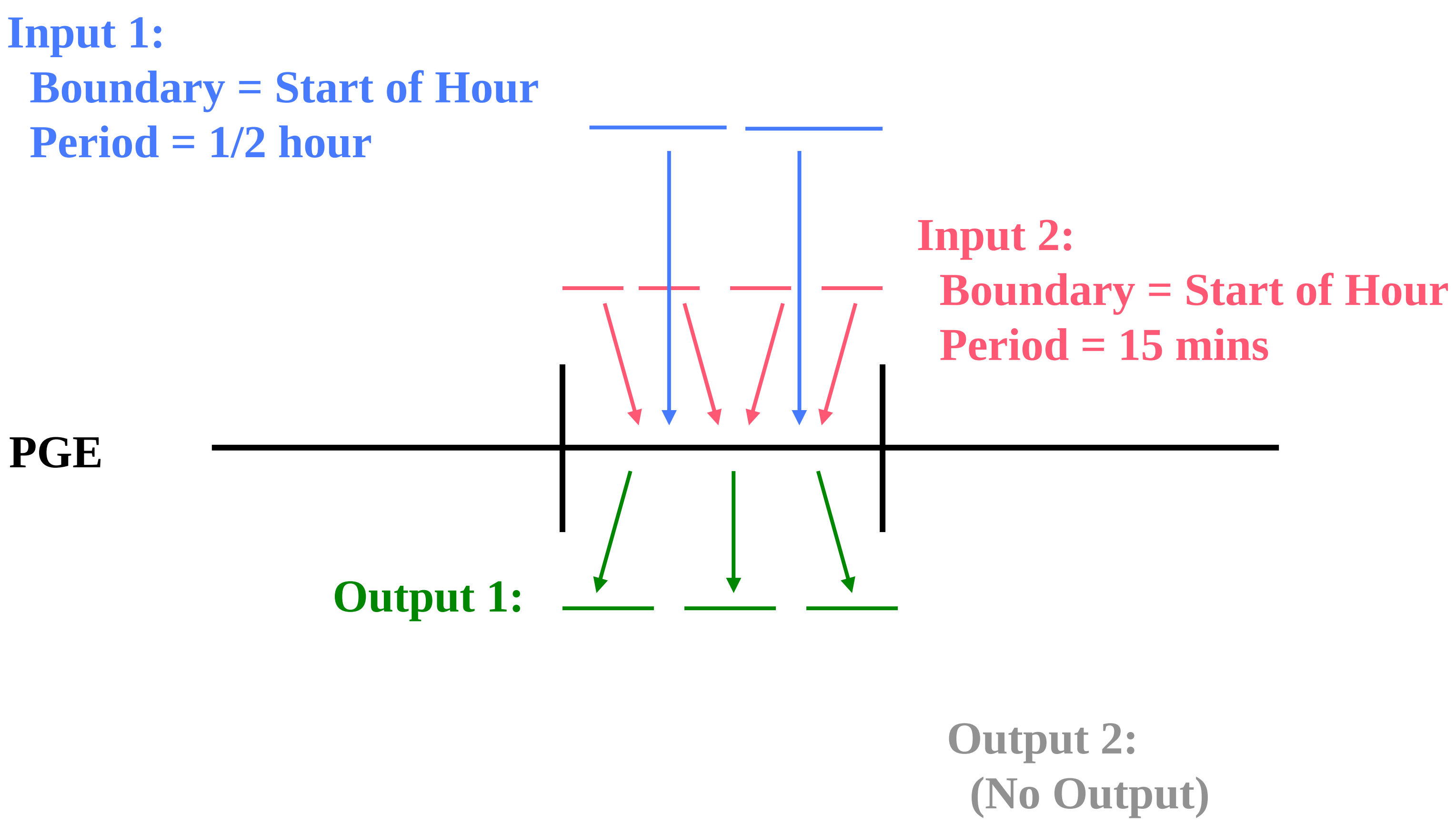
Production Requests (Cont.)

- Minimum/Maximum Number of Granules Production Rule (Cont.)
  - Different because it works for both input and output
  - PGE can request of a range of inputs (i.e., 1-10)
  - If a PGE needs at least three granules of an input…
    - Minimum number of granules is defined as three
    - PGE is not executed until at least three granules are available
  - Optional outputs are defined when the Minimum Number of Granules is set to zero
    - PGE can produce none of the particular type of output and still be considered successful
  - If a PGE has a non-zero value for a Minimum Number of Granules associated with an output, and fails to produce any granules of that output type, it is marked as failed

Production Requests (Cont.)

- Illustration of Minimum/Maximum Number of Granules Production Rule
  - PGE processes data related to a one-hour period and takes in both Input 1 and Input 2
  - Input 1 has a PERIOD of 1/2 hour
    - Every PGE run requires two Input 1 granules
  - Input 2 has a PERIOD of 15 minutes
    - There are four Input 2 granules for every PGE run
  - PGE produces three Output 1 granules for each run
  - In this case it does not produce any Output 2 granules

Example of Minimum/Maximum Number of Granules Rule



Production Requests (Cont.)

- **Illustration of Minimum/Maximum Number of Granules Production Rule (Cont.)**
  - **Minimum and maximum values can affect each input and output**
  - **Input 1:**
    - **If Minimum Granules is set to anything equal to or less than two for Input 1, the PGE is executed**
    - **If Minimum Granules is set to three, the PGE is not scheduled (there are not enough Input 1 granules to make the minimum)**
    - **If Maximum Granules is set to anything equal to or greater than two for Input 1, the PGE is executed**
    - **If Maximum Granules is set to one, the PGE is not scheduled (there are too many Input 1 granules - the number exceeds the maximum that the PGE can process)**

Production Requests (Cont.)

- **Illustration of Minimum/Maximum Number of Granules Production Rule (Cont.)**
  - **Input 2:**
    - **If Minimum Granules is set to anything equal to or less than four for Input 2, the PGE is executed**
    - **If Minimum Granules is set to five, the PGE is not scheduled (there are not enough Input 2 granules to make the minimum)**
    - **If Maximum Granules is set to anything equal to or greater than four for Input 2, the PGE is executed**
    - **If Maximum Granules is set to three, the PGE is not scheduled (there are too many Input 2 granules - the number exceeds the maximum that the PGE can process)**

Production Requests (Cont.)

- **Illustration of Minimum/Maximum Number of Granules Production Rule (Cont.)**
  - **Output 1:**
    - **If Minimum Granules is set to anything equal to or less than three, the PGE executes successfully**
    - **If Minimum Granules is set to four, the PGE is marked as failed (did not produce the expected number of output granules)**
    - **If Maximum Granules is set to anything equal to or greater than three, the PGE executes successfully**
    - **If Maximum Granules is set to two, the PGE is marked failed (produced too many output granules)**
  - **Output 2:**
    - **If Minimum Granules is set to anything other than zero, the PGE is marked failed (did not produce the expected number of output granules)**
    - **If Maximum Granules is set to anything equal to or greater than zero for Output 2, PGE is successful**

Production Requests (Cont.)

- Optional DPRs Production Rule
  – Also called the Data-Scheduled Production Rule
  – Makes the execution of a PGE subject to the availability of a "key input"
  – System generates DPRs for every possible instance of the key input data but executes only the DPRs for which data are either produced in data processing or can be acquired from the archive
  – Applies to PGEs that process certain kinds of non-routine data
    - Routine Data: Data that can be predicted, that come in at specific intervals and are always of a specified length
    - Non-Routine Data: Data that cannot be predicted because they come in at random periods and/or their length is variable (Examples include an "optional" output of an upstream PGE, or data that are archived at random periods; e.g., some forms of ASTER data)

Production Requests (Cont.)

- Optional DPRs Production Rule (Cont.)
  - An Optional DPR has as its key input a non-routine data type
  - There are two sets of circumstances that lead to the scheduling of Optional DPRs:
    - Every possible time that the input is produced in data processing (i.e., the key input is produced as an "optional" output by an upstream PGE)
    - Whenever a new granule (of a particular data type) can be acquired from the archive (e.g., archived data that were inserted at unpredictable times)

Production Requests (Cont.)

- Optional DPRs Production Rule (Cont.)
  - Example 1: Key input is produced as an "optional" output by an upstream PGE
    - One MODIS PGE produces a certain product only when the input data were collected during the satellite's "Day" mode
    - A second MODIS PGE is scheduled to use the optional ("Day"-mode) product from the first MODIS PGE as its key input
    - Second MODIS PGE is scheduled to run after every instance of the first MODIS PGE
    - Only the DPRs that can use the optional products resulting from runs of the first MODIS PGE are executed
    - The remaining DPRs cannot be executed because there is no input data for them

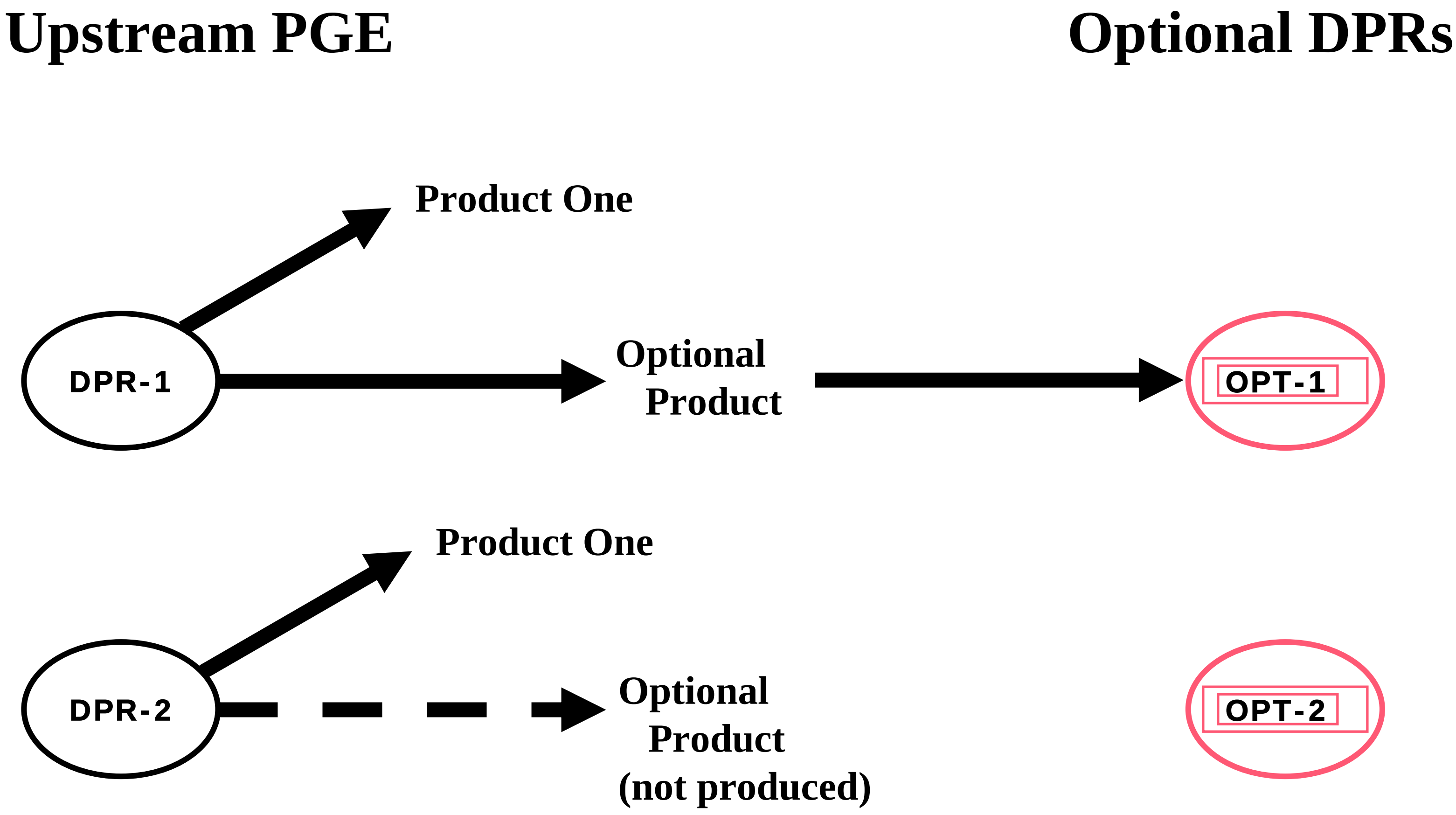
Production Requests (Cont.)

- Optional DPRs Production Rule (Cont.)
  - Example 2: Key input granules were inserted into the archive at unpredictable times
    - ASTER routine processing
    - DAAC ingests and archives ASTER production data from tapes supplied by the ASTER Ground Data System on a frequent but not predictable basis
    - When creating a Production Request, the Production Planner specifies the insertion time range as opposed to the collection time
    - Insertion time range = time period when the desired data were archived
    - Collection time = when the satellite instrument gathered the data
    - DPRs specifying the ASTER PGE are scheduled and executed for the data granules that were actually inserted in the archive during the specified time range

Production Requests (Cont.)

- Illustration of the Optional DPRs Production Rule
  - Two DPRs (i.e., DPR-1 and DPR-2) for the upstream PGE
  - Two DPRs (i.e., OPT-1 and OPT-2) for the PGE subject to the Optional DPRs Production Rule
  - "Optional DPRs" PGE takes as input the optional output of the upstream PGE
  - When it is executed, DPR-1 produces the optional output, so the dependent DPR (OPT-1) is executed
  - DPR-2 (on which OPT-2 depends) does not produce the optional output so OPT-2 is not executed

Example of Optional DPRs Production Rule



Production Requests (Cont.)

- Metadata Checks and Metadata Query Production Rules
  – Similar in definition and use
  – Both allow the PGE to specify granule-level metadata values that define whether the PGE can accept one (or more) of its inputs
  – Difference is in the results of metadata search

Production Requests (Cont.)

- **Metadata Checks and Metadata Query Production Rules (Cont.)**
  - **Metadata Checks Production Rule**
    - When PLS requests the Science Data Server to search for the input(s), the Science Data Server "checks" the metadata of all granules that match the time frame with respect to the allowed value(s)
    - If any granule fails to match the specified value(s), the PGE is not executed
  - **Metadata Query Production Rule**
    - When PLS requests the Science Data Server to search for the input(s), the Science Data Server adds to the query the PGE's desired metadata value(s)
    - Only the granules that match the time frame of the PGE plus the granule-level metadata value(s) specified by the PGE are staged as input to the PGE
    - If no granules are found matching the conditions and the input is not optional, the PGE is not executed

Production Requests (Cont.)

- **Metadata Checks and Metadata Query Production Rules (Cont.)**
  - **Example of Metadata Checks:**
    - **A MODIS PGE is run when the Percent Cloud Cover of its inputs is greater than 25 percent**
    - **The Metadata Checks Production Rule is used to specify the granule-level metadata value of greater than 25**
    - **When the PGE is scheduled and is ready to start, two granules match the timeframe of the Production Request for the input with the Metadata Check**
    - **If both granules have a Percent Cloud Cover greater than 25 percent, execution of the PGE starts and both granules are staged**
    - **If one of the granules has a Percent Cloud Cover of 15 percent, the PGE is not executed**

Production Requests (Cont.)

- **Metadata Checks and Metadata Query Production Rules (Cont.)**
  - **Example of Metadata Query:**
    - **A MODIS PGE is run when as many granules as possible of one of its inputs have a QA Value = "Good"**
    - **The Metadata Query Production Rule is used to specify the granule-level metadata value = "Good"**
    - **When the PGE is scheduled and is ready to start, two granules match the time frame of the production request for the input with the Metadata Query**
    - **If both granules have a QA Value = "Good", execution of the PGE starts and both granules are staged**
    - **If one of the granules has a QA Value = "Bad", the PGE executes but with only one granule (the one with QA Value = "Good")**

Production Requests (Cont.)

- **Metadata Checks and Metadata Query Production Rules (Cont.)**
  - **Metadata Checks and Metadata Query Production Rules are used in conjunction with the times specified in the Basic Temporal Production Rule or other production rules**
  - **Metadata Check or Query is added information that further refines what granules are sought by the PGE**

Production Requests (Cont.)

- **Metadata Checks and Metadata Query Production Rules (Cont.)**
  - **For past data**
    - **Production Request Editor performs the Metadata Query or Metadata Check immediately**
    - **Depending on what other production rules may be applicable, if no data is found to match the “query” (or data is found that does not match the “check”), the DPR fails to be created**
  - **For future data**
    - **Metadata Query/Metadata Check is put off until the DPR Data Collection Stop Time passes plus the value defined in the ODL for QUERY_DELAY**
    - **Delay allows the Metadata Query/Metadata Check to be put off until it's likely that all matching data will be present**

Production Requests (Cont.)

- **Metadata Checks and Metadata Query Production Rules (Cont.)**
  - **Multi-Granule ESDTs are a special case of the Metadata Query Production Rule**
    - **Used for PGE inputs or outputs when more than one granule of the same ESDT exists for the same temporal range (time period)**
    - **Multi-Granule ESDT mechanism employs a metadata parameter to differentiate between "equal in time" granules**

Production Requests (Cont.)

- **Metadata Checks and Metadata Query Production Rules (Cont.)**
  - **Data Day is an addition to the Metadata Query Production Rule**
    - **Involves runtime parameter values**
    - **Two settings (i.e., “Start Data Day” and “End Data Day”) allow a PGE to perform a Metadata Query for the start of the Data Day and the end of the Data Day**
    - **The Start Data Day and End Data Day are calculated by subtracting twelve (12) hours from the starting day for which the PGE is executing and adding twelve (12) hours onto the ending day for which the PGE is running. So if the PGE is running from 00:00:00 on 07/04 to 00:00:00 07/05 then the Start Data Day = 07/03 12:00:00 and the End Data Day = 07/06 12:00:00.**

Production Requests (Cont.)

- Illustration of the Metadata Checks and Metadata Query Production Rules
  - If no Metadata Check or Query were applicable, a particular PGE would use three granules of input (i.e., Granules A through C)
  - However, in the illustration the metadata value %CloudCover is to be checked/queried
    - Granules A through C each have a different value for %CloudCover
  - There can be more than one Metadata Check or Metadata Query on a given input
    - A Metadata Check on %CloudCover can be combined with a Metadata Query on another parameter to further limit the input

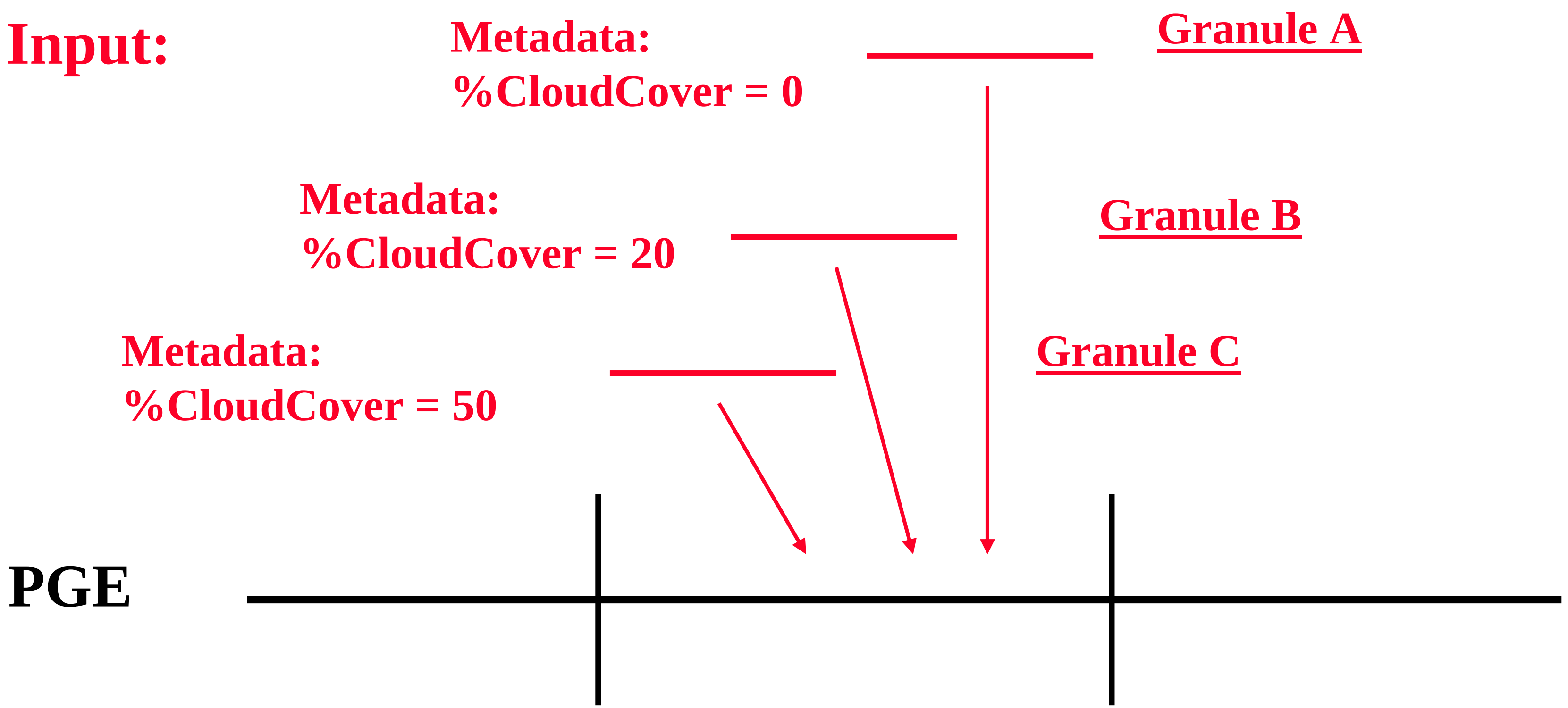
Production Requests (Cont.)

- **Illustration of the Metadata Checks and Metadata Query Production Rules (Cont.)**
  – **The following results demonstrate differences between Metadata Checks and Metadata Query:**
    - **Metadata Check of %CloudCover < 80: All three granules are acquired and the PGE is executed**
    - **Metadata Query of %CloudCover < 80: All three granules are acquired and the PGE is executed**
    - **Metadata Check of %CloudCover = 50: PGE is not scheduled because only one of the three granules (Granule C) meets the criterion**
    - **Metadata Query of %CloudCover = 50: Granule C is found and if the PGE's Min/Max Granules parameters are set to allow one granule, that one granule is acquired and the PGE is executed**

Production Requests (Cont.)

- **Illustration of the Metadata Checks and Metadata Query Production Rules (Cont.)**
  - **The following results demonstrate differences between Metadata Checks and Metadata Query:**
    - **Metadata Check of %CloudCover = 20: PGE is not scheduled because only one of the three granules (Granule B) meets the criterion**
    - **Metadata Query of %CloudCover = 20: Granule B is found and if the PGE's Min/Max Granules parameters are set to allow one granule, the granule is acquired and the PGE is executed**
    - **Metadata Check of %CloudCover < 20: PGE is not scheduled because only one of the three granules (Granule A) meets the criterion**
    - **Metadata Query of %CloudCover < 20: Granule C is found and if the PGE's Min/Max Granules parameters are set to allow one granule, the granule is acquired and the PGE is executed**

Example of Metadata Checks/Query



Production Requests (Cont.)

- Spatial Query Production Rule
  - PGE selects input(s) based on the spatial coverage of another input (called the key input)
  - PDPS queries the Science Data Server for the spatial coverage of the key input
  - PDPS uses key input's spatial coverage in acquiring any subsequent inputs that the PGE has requested that have the same spatial coverage
  - Without specifying coordinates, PDPS can match inputs against the spatial constraint of the key input, and give to a PGE only those granules which overlap in area

Production Requests (Cont.)

- **Spatial Query Production Rule (Cont.)**
  - **Example:**
    - **Level 0 input data for an ASTER DPR covers a small section of the Earth**
    - **The PGE requires ancillary data that covers the same area to complete its processing**
    - **The PGE uses the Spatial Query Production Rule to mark the geographic input as its key input**
    - **The PGE specifies that the ancillary input is to be retrieved for the same spatial coverage as that of the key input**
    - **When PDPS finds an input granule for the PGE, it performs a Spatial Query to acquire the ancillary input with the same spatial coverage as that of the key input**

Production Requests (Cont.)

- **Spatial Query Production Rule (Cont.)**
  - **Spatial Pad is an addition to the Spatial Query Production Rule**
    - **Spatial Pad is a means of padding the spatial constraints of the key input**
    - **The specified pad is added to all sides of the key input's spatial shape**
    - **All granules that intersect the expanded area are retrieved**

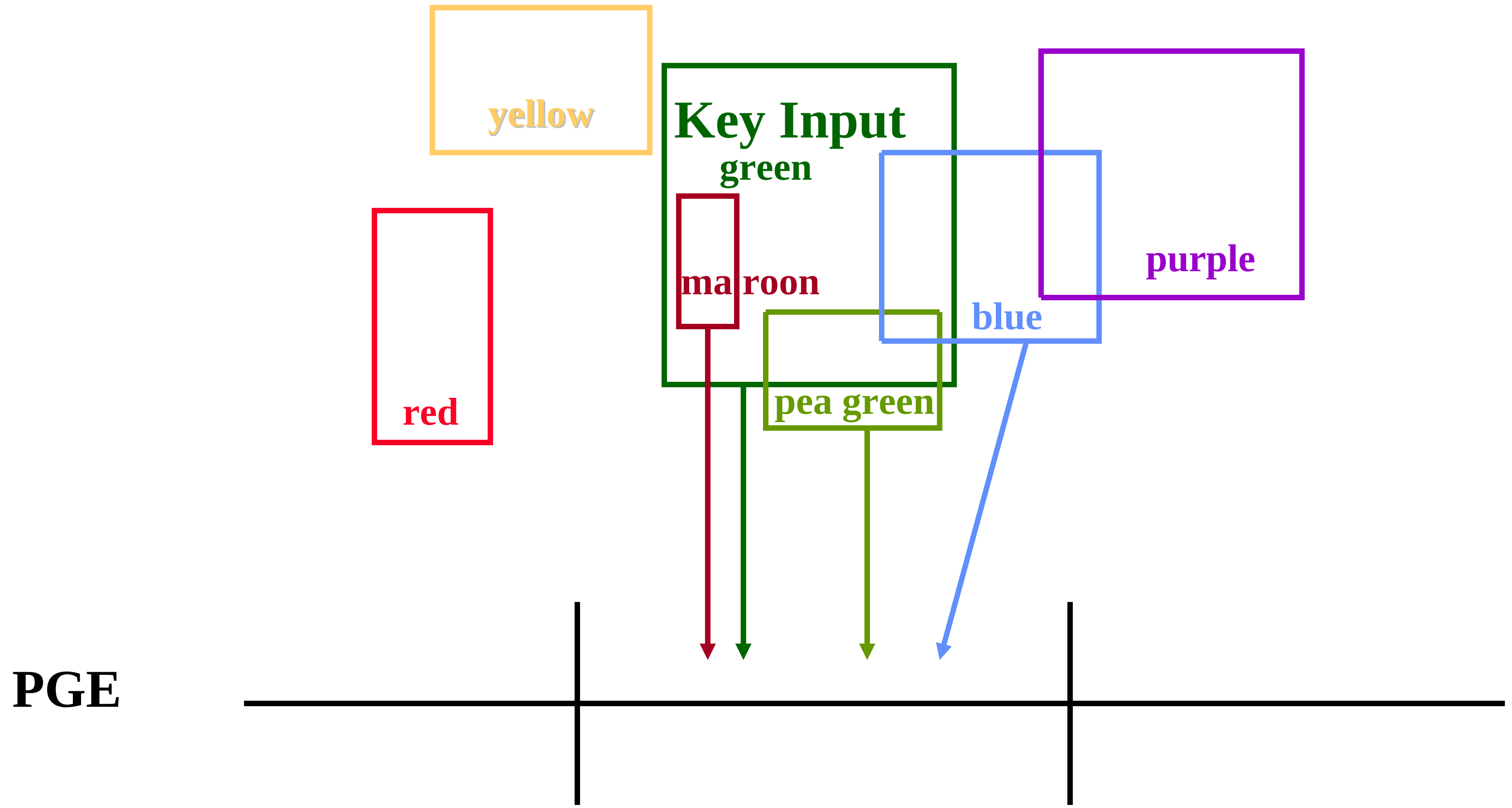
Production Requests (Cont.)

- Illustration of the Spatial Query Production Rule
  - PGE has two input types
    - One is the key input
    - The other type of input has granules labeled with the names of various colors
  - One granule (i.e., “green”) of the key input is found
  - Spatial coordinates of the granule are retrieved
  - All inputs of the second ESDT are checked for overlap with the key input’s coordinates

Production Requests (Cont.)

- **Illustration of the Spatial Query Production Rule (Cont.)**
  - **Assuming that all granules relate to the same time period, the granules are evaluated as follows:**
    - **"yellow" granule is not retrieved as an input (spatial coordinates do not overlap with those of the key)**
    - **"red" granule is not retrieved as an input (same as "yellow")**
    - **"blue" granule is retrieved as an input (spatial coordinates overlap with those of the key input)**
    - **"maroon" granule is retrieved as an input (same as "blue")**
    - **"pea green" granule is retrieved as an input (same as "blue")**
    - **"purple" granule is not retrieved as an input (same as "yellow")**

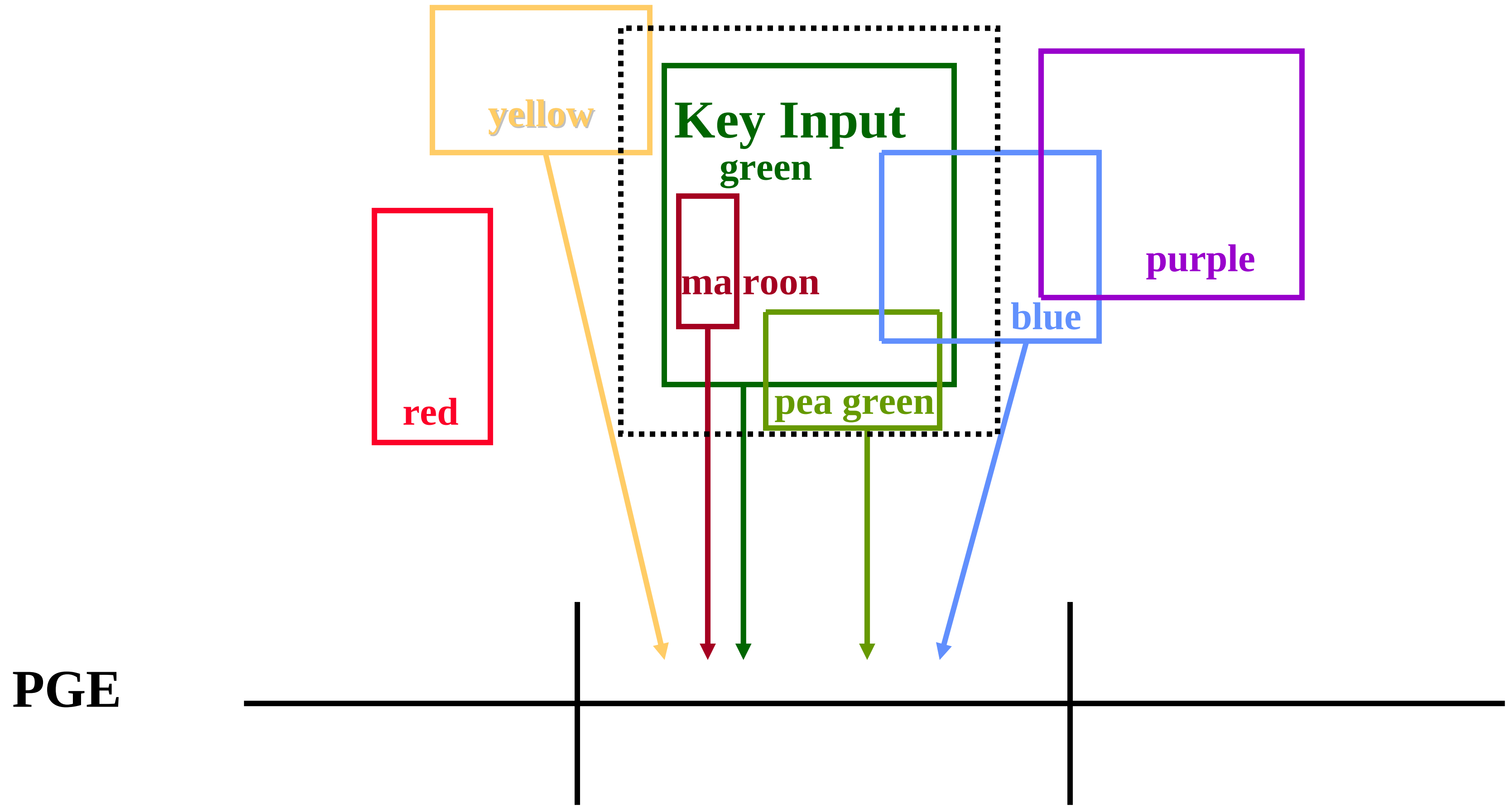
Example of Spatial Query



Production Requests (Cont.)

- **Illustration of Spatial Query with Spatial Pad**
  - Assuming that all granules relate to the same time period, the granules are evaluated as follows:
  - period, the granules are evaluated as follows:
    - "yellow" granule is retrieved as an input (spatial coordinates do not overlap with those of the key but do overlap the padded area)
    - "red" granule is not retrieved as an input (spatial coordinates do not overlap with either the key or pad)
    - "blue" granule is retrieved as an input (spatial coordinates overlap with the key)
    - "maroon" granule is retrieved as an input (same as "blue")
    - "pea green" granule is retrieved as an input (same as "blue")
    - "purple" granule is not retrieved as an input (same as "red")

Example of Spatial Query with Spatial Pad



Production Requests (Cont.)

- **Closest Granule Production Rule**
  - PGE requests the nearest input granule from the Data Processing Request time
  - PDPS requests a search forward or backward for a specified period of time until it finds a granule that matches the request; however, ...
    - Number of queries performed is limited
    - Period length of the query is limited
  - Supersedes the Most Recent Granule Production Rule
    - Allowed the search for inputs to go back only, not forward

Production Requests (Cont.)

- **Closest Granule Production Rule (Cont.)**
  - **Example:**
    - **A PGE processes data at daily intervals and could use a particular type of calibration granule that would allow it to determine the nearest parameters of the instrument**
    - **Although most calibration coefficients are defined as static granules, in this case there is a dynamic granule that is received about once a month**
    - **The closest such granule would be optimal, so the PGE uses the Closest Granule Production Rule to search forward or backward from the time of the DPR to find the nearest calibration granule**

Production Requests (Cont.)

- Closest Granule Production Rule (Cont.)
  - Three values determine the period of the query
    - Period (Offset)
    - Direction
    - Maximum Number of Queries

Production Requests (Cont.)

- **Closest Granule Production Rule (Cont.)**
  - Period (Offset)
    - Tells the PDPS software the query duration
  - Direction
    - Indicates whether the query goes forward (positive) or backward (negative) in time
    - In the PlDataTypeReq table in the PDPS database, the Direction and the Period information are combined, so the sign of value (+ or -) indicates the Direction and the magnitude indicates the Period
  - Maximum Number of Queries
    - Tells the PDPS software how many time periods (as defined by the Offset) to search (either forward or backward in time) for a matching granule

Production Requests (Cont.)

- Closest Granule Production Rule (Cont.)
  – PDPS does a Basic Temporal query before using Closest Granule to find the input
  – If the desired input is not found within the time period of the DPR, PDPS performs a query (in the specified direction) against the Science Data Server for the period defined by the offset
  – If no matching granule is found, PDPS repeats the query, going backward or forward in time by the value specified in the offset
  – If no acceptable granule has been found before the maximum number of queries is reached, PDPS fails to generate the DPR due to insufficient input data
  – special case for the forward search:
    - when the next search interval exceeds the current time, the search stops at the current time

Production Requests (Cont.)

- **Closest Granule Production Rule (Cont.)**
  - **If the DPR is planned for a future time, the DPR is created using place-holder granules and a timer is activated**
  - **When the current time reaches the stop time of DPR, the timer invokes the Closest Granule method to search for the actual granules, which are then used to replace the dummy granules**

Production Requests (Cont.)

- Illustration of the Closest Granule Production Rule
  - PGE boundary is “start of day”
  - PGE period is one hour
    - PGE is scheduled to run for one hour’s worth of input data
  - Input period is one hour
    - Can come in at any hour of the day
  - PGE requests one granule of input
  - Offset is 6
    - Queries in six-hour intervals
  - Direction is backward
  - Maximum number of queries is two

Production Requests (Cont.)

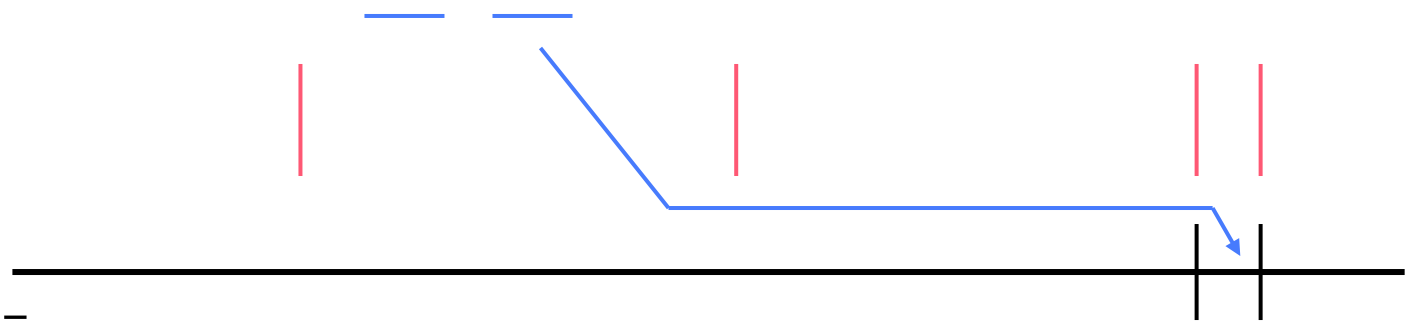
- Illustration of the Closest Granule Production Rule (Cont.)
  - PDPS performs a query for the input based on the time period of the DPR
    - No matching data is found
  - PDPS uses the Closest Granule information to query for a six-hour period beginning six hours before the start time of the DPR
    - Again nothing is found
  - A second Closest Granule query is performed, this one six hours before the last Closest Granule query
    - Two granules are found that match the query
  - PDPS selects the granule that is later in time and schedules the PGE to use it as input

Production Requests (Cont.)

- **Illustration of the Closest Granule Production Rule (Cont.)**
  - **If the Closest Granule Production Rule were used in conjunction with the Minimum/Maximum Number of Granules Production Rule, it might be possible for both granules to be selected in the previously described Closest Granule query**
    - **If the example included setting the Maximum Number of Granules to two, both granules would be selected as input to the PGE**

Example of Closest Granule

Input: Boundary = Start of Hour
Period = 1 hour



PGE:
Boundary =
Start of Day
Period =
1 hour

Closest Granule Query:
Query Period = -6 hours
Max Queries = 2

Production Requests (Cont.)

- Orbital Processing Production Rule
  – Uses the orbit of the spacecraft to determine the time period for the inputs and outputs of the PGE
  – Example:
    - A PGE processes Level 0 data related to each orbit of the Terra satellite
    - The Terra satellite has an orbital period of 98 minutes so the PGE is scheduled to process data for each 98-minute interval
    - Since Level 0 data are received every two hours, the data staged for the PGE include every Level 0 granule that falls within the 98-minute PGE interval
    - Only one granule of Level 0 data is relevant to some 98-minute orbits
    - Two granules of Level 0 data are relevant to other 98-minute orbits

Production Requests (Cont.)

- **Orbital Processing Production Rule (Cont.)**
  - Uses the “period” and “boundary” concepts just like the Basic Temporal Production Rule
    - Orbit of the spacecraft is taken into account when a PGE or its data are marked as orbit scheduled
  - When responding to a Production Request for orbit-scheduled processing, PDPS determines the orbit of the satellite via information from SSI&T
  - Information in the PDPS database gives the start time and length of a particular orbit or set of orbits
  - PDPS extrapolates (or interpolates in the case of an orbit between two orbital periods stored in the database) the start and end times of the PGE that is specified in the Production Request
  - Data are sought on the basis of the derived start and stop times and the appropriate data granule(s) is/are staged before the PGE is executed

Production Requests (Cont.)

- Orbital Processing Production Rule (Cont.)
  - Orbit model is a model of the satellite's orbits that allows PDPS to perform extrapolations for the Orbital Processing Production Rule
    - Model is a combination of a database table and a simple algorithm
    - Database table stores Orbit Number/Orbit Start Time/Orbital Period combinations
    - Algorithm uses the data to compute the same type of data relevant to subsequent orbits
    - Works by extrapolation (unable to calculate data for any orbit that precedes the earliest entry in the database table)
    - Data for the Orbit Model is specified in the Orbit Model ODL file, which is read only if the PGE requires orbital information

Production Requests (Cont.)

- Orbital Processing Production Rule (Cont.)
  - Orbital path is the path of the satellite over the Earth
    - Number from 0-233 that indicates the region of the Earth covered by a particular orbit
    - Because of the implementation of Orbital Path, there needs to be a mapping between the orbital path calculated by PDPS and the orbital path number expected by the PGEs
  - Runtime parameters can be set to values associated with Orbital Processing
  - The following orbital parameters can be placed under runtime parameters:
    - Orbit Number
    - Orbital Path Number
    - Orbit Number within the Day
    - Granule Number within the Orbit

Production Requests (Cont.)

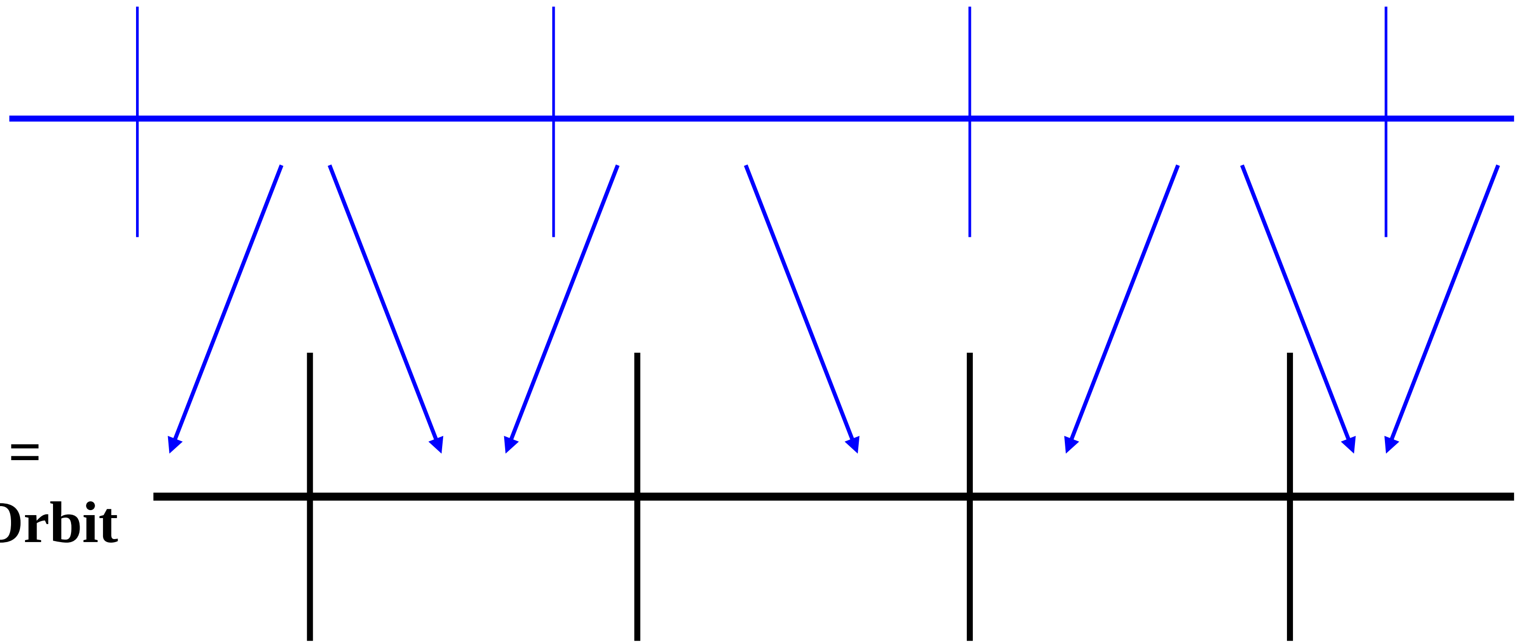
- **Orbital Processing Production Rule (Cont.)**
  - Orbit Number
    - Number of the orbit (starting from zero) and continually increasing
  - Orbital Path Number
    - Number of the path that maps to the orbit number
    - Orbital path number is the 0-233 orbital path traversed by the satellite
  - Orbit Number within the Day
    - Number of the orbit within the given day
    - Includes any orbit that starts within the given day
  - Granule Number within the Orbit
    - Number of the granule within a given orbit
    - Includes any granule that starts within the given orbit

Production Requests (Cont.)

- Illustration of the Orbital Processing Production Rule
  – PGE takes a two-hour input, but is scheduled based on the orbit time and period of the satellite
  – PDPS uses the data collected at SSI&T to predict the time of the orbit and performs the query to the Science Data Server for the input based on that extrapolated or interpolated orbital time
  – Granules of input data are allocated to DPRs based on their ability to cover the DPR’s time period
    - Length of an orbit is less than the period of the two-hour input
    - Sometimes a single granule may cover the input time range of a PGE execution
    - At other times two granules are required
    - Rule would work equally well if the data were of a shorter period than the orbit of a satellite

Example of the Orbital Processing Production Rule

Input: Boundary = Start of Hour
Period = 2 hours



PGE:
Boundary =
Start of Orbit
Period =
1 orbit

Production Requests (Cont.)

- **Multiple DPRs for Insertion Time Production Rule**
  - **Allows the creation of DPRs for multiple granules with the same insertion time**
    - **Affects ASTER L1B routine processing only**
  - **Is implemented when the Production Planner creates a production request**
    - **On the Production Request Editor the Production Planner enters the Duration information (Begin and End date and time) for the Insertion Time (versus Collection Time) and then ensures that the Multiple DPRs toggle button is depressed**
    - **If the Multiple DPRs toggle button is not depressed, the default rule is applied and when the production request is saved a DPR is created for the most recent granule with the specified insertion time**

Production Requests (Cont.)

- Multiple DPRs for Insertion Time Production Rule
  – Applies to the key input (e.g., ASTER L1B) only
    - It does not affect any ancillary data types (for which the most recent granule is selected)
  – When the Multiple DPRs for Insertion Time production rule is invoked, a DPR is generated for each key-data-type granule returned from the insertion-time query
    - Therefore, if multiple key-data-type granules are returned for the same collection period, multiple DPRs are generated
    - The DPR IDs are different in the last three characters, which are generated randomly

Production Requests (Cont.)

- **Tiling Production Rule**
  – **PGE is run for a series of specific geographic locations called "tiles"**
  – **Tiles are defined before the PGE is scheduled, specifying the longitude and latitude of four points that outline each tile**
  – **When the PGE is scheduled, it is scheduled for an entire day**
    - **Data is queried based on both a time frame and the geographic location specified**
  – **Each run of the PGE for the day is for a specific tile, and only data that overlap or fit within the geographical coordinates of the tile are staged for the PGE**

Production Requests (Cont.)

- **Tiling Production Rule (Cont.)**
  - **Example:**
    - A MODIS PGE is designed to run on data for a specific geographic location every day
    - The location is expressed as a polygon defined by latitude and longitude coordinates
    - The MODIS PGE is scheduled every day, and data are retrieved that match the time period (the day for which the PGE is being executed) and some part of it falls within the geographic constraints of the tile
    - The PGE runs and produces data that define information about the particular tile
  - **Period and boundary are used to specify the timing of input data and provide indications of how often the PGE should be executed**
  - **At least some of the input data are retrieved on the basis of the coordinates defined for the tile on which the PGE is executing**

Production Requests (Cont.)

- **Tiling Production Rule (Cont.)**
  - **There are really two kinds of tiling:**
    - **PGE takes in data based on geographic shapes (tiles) and produces an output or outputs for the specified geographical coverage**
    - **PGE takes in an already tiled product as input (This form of tiling is more like a Metadata Query using a runtime parameter value to acquire the correct tiled data)**
  - **Runtime parameters can be set to the ID of the tile being processed**
    - **Since PDPS schedules a Tiling PGE to run once per tile, it can pass the identifier of the tile to the PGE**
    - **The identifier can be placed under a specified runtime parameter in the PCF, or it can be used in a Metadata Query for a PGE that would use already tiled data as input**

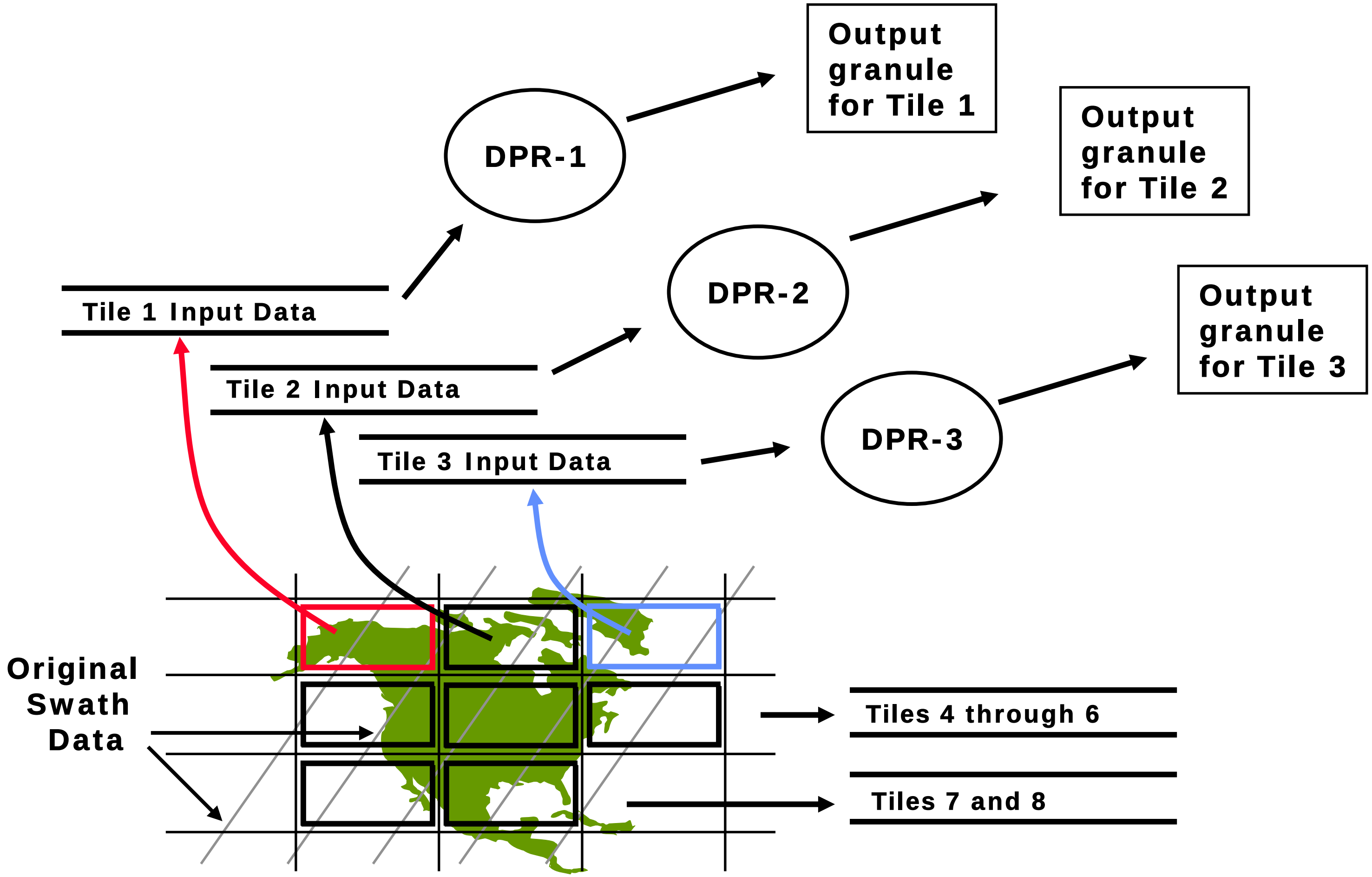
Production Requests (Cont.)

- Illustration of the Tiling Production Rule
  - PGE runs once per defined tile
  - For every tile in the Tile Scheme a Data Processing Request is created to run using data that match the geographic extent of the tile
  - PDPS sends the coordinates of the tiles (e.g., Tiles 1 through 3) to the Science Data Server when requesting data
  - Only the granules that fall fully or partially within the defined tile are acquired
  - The PGE itself must be set up to handle the fact that the entire area of the tile may not be covered by available data

Production Requests (Cont.)

- **Illustration of the Tiling Production Rule (Cont.)**
  - **PGE must set the metadata of the output products so a downstream Tiling PGE can acquire the correct granules for a given tile**
    - **PDPS does not keep track of tiles once they have been produced**
  - **PDPS matches up the granules needed for a downstream PGE via a query to the Data Server Subsystem**

Example of Tiling



Production Requests (Cont.)

- **Possible future enhancements to the Tiling Production Rule:**
  - **Zonal Tiling**
    - **Supports tiles that cover a band around the Earth between two given latitudes**
  - **Tile Clustering**
    - **Involves grouping tiles that cover nearby geographic locations together so that data that span the tiles may be staged only once**
    - **Intended to improve the performance of Tiling**
    - **Also provides for the ability to prioritize one group of tiles over others (so specific geographic outputs are produced before other geographic outputs)**

Production Requests (Cont.)

- Production Planning Considerations
  - Production Request (PR) is a template request to generate a particular data product
    - results in a production run of the associated SCF-provided PGE
  - PR specifies a range over which the data products are to be produced or the PGEs are to be scheduled
    - time
    - snapshot
    - data
    - orbit

Production Requests (Cont.)

- **Production Planning Considerations (Cont.)**
  - **During normal operations Production Planner should not have to add PRs to the PDPS database very frequently**
    - **PR might request that the data product be produced for only a single day's data**
    - **PR might request that data products be produced for every opportunity of input data for several months**
    - **Early in a mission the SCF may prefer to request processing for a short time period only**
    - **SCF reviews the quality of the products and notifies the DAAC of the need for any changes to the PR**
    - **When the SCF has developed a good understanding of the instrument's behavior, the team may be comfortable requesting processing for months at a time**

Production Requests (Cont.)

- **Production Planning Considerations (Cont.)**
  – **DAAC operations may have operational reasons for wanting to issue processing requests for a limited time period**
  – **Production Planner has to balance the various considerations when determining whether or not to create or update a PR**
  – **Planning decisions are made on the basis of locally defined planning strategies for supporting the SCFs' data processing needs**
  – **Production Planner must coordinate with the Resource Planner**
    - **before planning production**
    - **resolve all resource allocation issues**
    - **determine what resources are available for use in processing**

Production Requests (Cont.)

- Chain Heads
  - A chain is a related set of DPRs in which the output of the first DPR in the chain (i.e., the “chain head”) is used as input for at least one subsequent DPR
    - Output of the subsequent DPR(s) may be used as input for further DPR(s) and so on until the other end(s) of the chain is (are) reached
    - Chained DPRs must be processed in the proper sequence from the chain head to the chain’s other end(s)
  - By default all DPRs in a chain are executed on a single computer
    - Most of the data for the chain are produced and consumed locally, so communication of data on network-mounted file systems is kept to a minimum

Production Requests (Cont.)

- **Chain Heads (Cont.)**
  - Production Request Editor provides a means by which the Production Planner can identify a PR as the first in a chain of PRs
  - The outputs of the DPR(s) that PLS creates from the chain-head PR are used as inputs to one or more subsequent DPR(s) specified in other PRs
    - Any downstream DPR (i.e., that requires the output of a preceding DPR as its input) is known as a "child" DPR
    - Any DPR that provides output for use as an input to a child DPR is a "parent" DPR (chain heads are parent DPRs)
    - To completely deactivate chain processing, declare every PGE to be a chain head
    - To enable total chain processing, declare no PGE to be a chain head

Production Requests (Cont.)

- **Chain Heads (Cont.)**
  - **The "Determine Chain" script (i.e., EcPlDetermineChain.pl) finds chains starting with the DPREP PGEs**
  - **In addition to designating chain heads in a PR, the Production Planner may select a particular virtual computer for running the chain-head DPRs**
    - **Selection of a virtual computer is optional**
    - **If no machine is specified, the system tries to schedule the DPR on the machine where the bulk of its accepted inputs (both static and dynamic inputs) are staged**

Production Requests (Cont.)

- Chain Heads (Cont.)
  - Implementation of chain heads includes ….
    - Addition of chainFlag and scheduledMachine columns to the PlDataProcessingRequest and PlProductionRequest database tables in the PDPS database
    - Addition of chainId column to the PlDataProcessingRequest database table in the PDPS database
    - The Production Request Editor sets the chainFlag and scheduledMachine values based on Production Planner input when a PR is created
    - It is also possible to set the chainFlag value using the Planning Workbench GUI at plan activation time for DPRs that have no "parent" DPRs producing data used for their input

Production Requests (Cont.)

- **Chain Heads (Cont.)**
  - **The Planning Workbench GUI calls the "Determine Chain" script when a plan is activated**
    - **/usr/ecs/*MODE*/CUSTOM/bin/PLS/EcPlDetermineChain.pl**
    - **At that time, chainFlag values for previously unflagged chain heads can be set and a chainId can be assigned to each DPR**
    - **If the Determine Chain script fails for some reason, the plan is not activated**

Production Requests (Cont.)

- **Intermittent Activation**
  - **PGE is set up to run on every *n*th instance of input data**
  - **To implement Intermittent Activation the Production Planner supplies the following information (via the Production Request Editor) when creating a production request:**
    - **“Skip” field on the Production Request Editor: Number of DPRs to be skipped (not executed) is entered in the field**
    - **“Keep” field on the Production Request Editor: Number of DPRs to keep (after skipping the specified number of DPRs) is entered in the field**
    - **“Skip First” button on the Production Request Editor: Selected to skip the first DPR (not selected if the first DPR is to be run)**

Production Requests (Cont.)

- Intermittent Activation (Cont.)
  - Planning Subsystem uses the “Intermittent DPR” information from the PR to establish a pattern of execution
  - Pattern is effective for the single PR in which the “number to skip” and the “number to keep” are specified
    - Pattern is not maintained between PRs

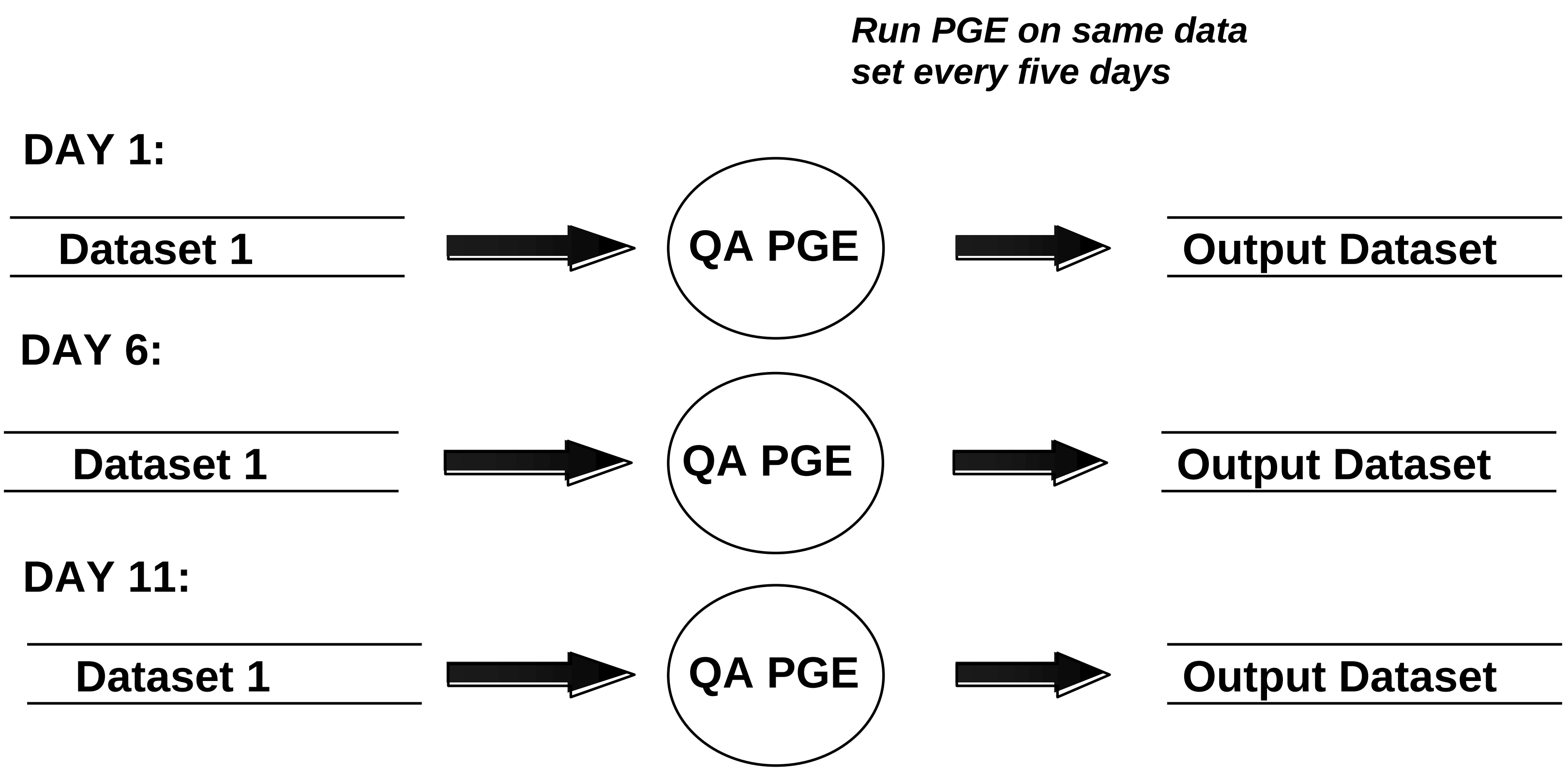
Production Requests (Cont.)

- **Illustration of Intermittent Activation**
  - Production Planner prepares a production request for a 14-day period, generating 14 DPRs
  - Production Planner made the following selections on the Production Request Editor:
    - Entered “4” in the Number to Skip field
    - Entered “1” in the Number to Keep field
    - Did not select the Skip First button

Production Requests (Cont.)

- Illustration of Intermittent Activation (Cont.)
  - The following results are obtained:
    - First DPR runs
    - Four DPRs (second through fifth) are skipped
    - Sixth DPR runs
    - Four DPRs (seventh through tenth) are skipped
    - Eleventh DPR runs
    - Remaining three DPRs (twelfth through fourteenth) are skipped

Example of Intermittent Execution



Production Requests (Cont.)

- Data Preprocessing (DPREP)
  - DPREP (data preprocessing) consists of sets of PGEs that use a statistical approach to convert Level 0 (L0) ephemeris and attitude ancillary data for a particular satellite (e.g., Terra, Aqua, or Aura) into SDP Toolkit native binary format and HDF format without altering or modifying the scientific content of the granules
  - Ephemeris data provide the following types of information:
    - Spacecraft location: ephemeris (or orbit) data include: latitude, longitude, and height
  - Attitude data provide the following types of information:
    - Orientation of the satellite, including yaw, pitch, and roll angles; and angular rates about the three axes

Production Requests (Cont.)

- Data Preprocessing (DPREP) (Cont.)
  - DPREP PGEs are supplied by the EMD Project, unlike most PGEs
    - Most PGEs are provided by the Science Computing Facilities that EMD supports
  - DPREP supports Terra, Aqua, and Aura operations

Production Requests (Cont.)

- **Terra DPREP**
  - **Three PGEs:**
    - **EcDpPrAm1EdosEphAttDPREP_PGE**
    - **EcDpPrAm1FddAttitudeDPREP_PGE**
    - **EcDpPrAm1FddEphemerisDPREP_PGE**
  - **Operationally, EcDpPrAm1EdosEphAttDPREP_PGE and EcDpPrAm1FddAttitudeDPREP_PGE are scheduled daily and run independently of one another**
  - **EcDpPrAm1FddEphemerisDPREP_PGE is scheduled and run on an as-needed basis**

Production Requests (Cont.)

- Terra DPREP (Cont.)
  - The first PGE (EcDpPrAm1EdosEphAttDPREP_PGE) produces both ephemeris and attitude data
    - However, the attitude data from the first PGE is not sure to be of good enough quality for science data processing
  - The second PGE (EcDpPrAm1FddAttitudeDPREP_PGE) processes attitude data from FDD that is of good enough quality for science data processing
  - If the data quality analysis performed by the first PGE indicates that the quality of the ephemeris from the spacecraft is poor, DPREP initiates the request for replacement ephemeris data from FDD
  - The third PGE (EcDpPrAm1FddEphemerisDPREP_PGE) processes the replacement ephemeris

Production Requests (Cont.)

- **Terra DPREP (Cont.)**
  - **Sources of information on the Terra DPREP PGEs and how to run them:**
    - **500-EMD-001, *Terra Spacecraft Ephemeris and Attitude Data Preprocessing***
    - **611-EMD-001, *Mission Operation Procedures for the EMD Project*, Chapter 26**
    - **Two files installed on the science processor hosts (e.g., e0spg11, g0spg11, or l0spg11) in the /usr/ecs/*MODE*/CUSTOM/data/DPS directory; i.e., “AM1_DPREP_README” and “HowToRunAm1DPREP”**

Production Requests (Cont.)

- **Terra DPREP (Cont.)**
  - The DPREP PGEs process Level Zero (L0) Terra (AM-1) spacecraft data (e.g., ESDT AM1ANC) provided by EDOS
  - In addition, the Terra DPREP PGEs use Terra FDD Attitude (AM1ATTF) data and may use Terra FDD Ephemeris (AM1EPHF) data
  - The output files/granules of the DPREP PGEs are subsequently used in the processing of science data from various instruments on the satellite
  - DPREP output granules provide valuable data concerning the location and orientation of the satellite when the data from the satellite's instruments were collected

Production Requests (Cont.)

- **Terra DPREP (Cont.)**
  - **DPREP processing is granule-oriented**
    - **The processing interval selects data granules from the archive for DPREP to process**
    - **Then the granules get processed to completion**
    - **All data (i.e., ephemeris and attitude data) that are processed by Terra DPREP arrive in two-hour segments**
    - **Therefore, processing intervals are selected in multiples of two hours**

Production Requests (Cont.)

- **Terra DPREP (Cont.)**
  - QA analysis is an important function of DPREP
    - It includes checking the continuity of the ephemeris and attitude data streams across data segments; i.e., the segment that is being processed and the immediately preceding and following segments
    - DPREP does this by performing consistency, limit, and data gap checks that bridge segment boundaries
    - The checks are initiated using data from the end of the immediately preceding segment and completed using data from the immediately following segment
    - When performing QA analysis on records close to a granule boundary, the QA window extends into the preceding or following granule as circumstances dictate
    - Consequently, Terra DPREP needs nominal access to the granules that immediately precede and follow the “current” granule

Production Requests (Cont.)

- **Terra DPREP: Ephemeris Processing**
  - EcDpPrAm1EdosEphAttDPREP_PGE consists of a script that coordinates the following three DPREP executables:
    - EcDpPrAm1EdosAncillary
    - EcDpPrAm1EdosEphemerisRepair
    - EcDpPrAm1ToolkitToHdf
  - EDOS L0 Ancillary processing must run at least two hours behind “real time”
    - DPREP performs consistency checks across granule boundaries and requires the L0 ancillary granule that follows the “current” L0 ancillary granule in order to perform the consistency check

Production Requests (Cont.)

- **Terra DPREP: Ephemeris Processing (Cont.)**
  - EcDpPrAm1EdosAncillary reads in the “current” EDOS L0 Ancillary granule (ESDT AM1ANC) and the next AM1ANC granule
  - EcDpPrAm1EdosAncillary reads in ephemeris and attitude data (ESDT AM1EPHN0 and ESDT AM1ATTN0) from the preceding run of the PGE to support QA
  - EcDpPrAm1EdosAncillary identifies data to be repaired in the ephemeris data stream
  - EcDpPrAm1EdosAncillary writes the ephemeris granules as temporary granules, not as Production Data Sets (PDSs)
  - If EcDpPrAm1EdosAncillary determines that the ephemeris data stream had repairable data quality problems, EcDpPrAm1EdosEphemerisRepair performs data repair on the temporary Toolkit-format ephemeris granule to produce the repaired granule

Production Requests (Cont.)

- **Terra DPREP: Ephemeris Processing (Cont.)**
  – **EcDpPrAm1ToolkitToHdf completes the process by generating an HDF-format EDOS ephemeris from the repaired Toolkit-format EDOS ephemeris granule**
  – **The granules that EcDpPrAm1EdosEphemerisRepair and EcDpPrAm1ToolkitToHdf produce are written as Production Data Sets**
  – **Temporary granules do not get archived, but do remain in the run-time directory for a short period of time in case they need to be examined**
  – **EcDpPrAm1EdosEphAttDPREP_PGE produces Toolkit- and HDF-format attitude (ESDTs AM1ATTN0 and AM1ATTH0) and ephemeris (ESDTs AM1EPHN0 and AM1EPHH0) granules**

Production Requests (Cont.)

- Terra DPREP: Ephemeris Processing (Cont.)
  - EDOS-supplied ephemeris data are the primary source of ephemeris for Terra
  - EDOS-supplied attitude data are not the primary source of attitude for Terra
    - Attitude data supplied by the Flight Dynamics Division (FDD) are considered the primary source of Terra attitude
  - EcDpPrAm1EdosAncillary performs a full complement of data quality analyses on the EDOS ephemeris data
    - In contrast EDOS-supplied attitude data are subject to minimal quality checks and never undergo data repair because EDOS-supplied attitude data are not considered good enough for science data processing
    - EDOS attitude data are “use at own risk” data
    - The data recommended for science data processing are the FDD attitude data routinely preprocessed by EcDpPrAm1FddAttitudeDPREP_PGE

Production Requests (Cont.)

- **Terra DPREP: FDD Attitude Processing**
  - **FDD attitude processing must run at least two hours behind “real time”**
    - **DPREP performs consistency checks across granule boundaries and requires the FDD Attitude granule that follows the “current” FDD Attitude granule to perform the consistency check**
  - **EcDpPrAm1FddAttitudeDPREP_PGE**
    - **Reads in both the “current” FDD attitude granule (AM1ATTF) and the next FDD attitude granule**
    - **Reads in the attitude granule (AM1ATTNF) it produced with its last run**
    - **Output is native format attitude file (AM1ATTNF), HDF format attitude file (AM1ATTHF) and associated metadata files**

Production Requests (Cont.)

- **Terra DPREP: FDD Attitude Processing (Cont.)**
  - **Because the FDD attitude data are extensively preprocessed by FDD prior to processing by DPREP, DPREP performs minimal data quality checking on the FDD Attitude data stream (primarily to catch transmission errors)**
  - **DPREP does not perform data repair of FDD attitude data**

Production Requests (Cont.)

- Terra DPREP: Terra Data Repair
  - Data repair is performed on the EDOS-supplied ephemeris data stream only
  - EcDpPrAm1FddEphemerisDPREP_PGE
    - If EcDpPrAm1EdosEphAttDPREP_PGE finds too many missing data points in the ephemeris data (e.g., AM1EPHH0 and AM1EPHN0 granules have gaps of 58 records or more - or about 60 seconds), DPREP requests an FDD replacement granule (AM1EPHF) by submitting a subscription to the Spatial Subscription Server
    - When the FDD ephemeris granule has been ingested, EcDpPrAm1FddEphemerisDPREP_PGE preprocesses the FDD Ephemeris granule to produce the FDD Toolkit ephemeris replacement granule
    - In addition to the Toolkit-format (AM1EPHN0) FDD ephemeris granule an HDF-format (AM1EPHH0) FDD ephemeris granule is produced

Production Requests (Cont.)

- **Terra DPREP: Terra Data Repair (Cont.)**
  - The FDD Toolkit ephemeris replacement granule replaces a Toolkit-format EDOS ephemeris granule directly within the EDOS Toolkit ephemeris data stream
  - Consequently, FDD ephemeris processing does not generate an ESDT that stands apart from the Toolkit-format EDOS ephemeris ESDT, but instead produces a different “flavor” of ephemeris data within the Toolkit-format EDOS ephemeris data stream
  - Because FDD performs preprocessing on replacement data, DPREP performs minimal replacement ephemeris data quality checking (primarily to catch transmission errors)
  - DPREP does not perform data repair on replacement ephemeris

Production Requests (Cont.)

- **Terra DPREP: Profiles**
  - **DPREP processing has data requirements beyond the current data segment (data from the preceding and following segments are used for performing consistency checks)**
    - **There is no guarantee that data from the preceding and following segments will always be available because adjacent segments may be in different granules**
    - **Consequently, four data processing profiles have been developed for each of the DPREP PGEs to accommodate the various permutations of data availability**

Production Requests (Cont.)

- **Terra DPREP: Profiles (Cont.)**
  - **Profile 1 (preceding and following data available) is used for nominal DPREP operation**
    - **It is the profile of each DPREP step that is run on a routine basis**
    - **It is quite flexible in that it can often proceed even when there is no granule from the immediately preceding segment or from the following segment**
  - **Profile 2 (no preceding data, but following data is available) is the boot-up process used for initializing DPREP processing of the ephemeris and attitude data streams**
    - **After Profile 2 has been run on a data segment, Profile 1 assumes processing responsibility on all data segments thereafter until data dropout**

Production Requests (Cont.)

- **Terra DPREP: Profiles (Cont.)**
  - **Profile 3 (preceding data available, but no following data) processes the data segment that immediately precedes data dropout (terminates processing on the ephemeris and attitude data streams)**
  - **Profile 4 is used for processing isolated data segments (not likely to be scheduled operationally)**

Production Requests (Cont.)

- **Terra DPREP: Profiles (Cont.)**
  - In the big picture of the mission, DPREP processing on the very first data segment would use Profile 2 (boot-up)
  - The next data segments would be processed using Profile 1 (nominal) processes
  - The very last data segment of the mission could be processed using Profile 3
  - However, given the processing flexibility of Profile 1, neither the scheduling of Profile 3 or the scheduling of Profile 4 is envisioned operationally

Production Requests (Cont.)

- **Aqua DPREP**
  - **Aqua DPREP processes:**
    - **EcDpPrPm1FddEphemerisDPREP_PGE**
      FDD Ephemeris Processing
    - **EcDpPrPm1AttitudeDPREP_PGE**
      ECS Mission Operations Segment (EMOS) Attitude Processing
  - **Sources of information on the Aqua DPREP PGEs and how to run them:**
    - **500-EMD-002, *Aqua Spacecraft Ephemeris and Attitude Data Preprocessing***
    - **611-EMD-001, *Mission Operation Procedures for the EMD Project*, Chapter 26**
    - **File installed on the science processor hosts (e.g., e0spg11, g0spg11, or l0spg11) in the /usr/ecs/*MODE*/CUSTOM/data/DPS directory; i.e., "HowToRunPm1DPREP"**

Production Requests (Cont.)

- Aqua DPREP (Cont.)
  - Includes the processing of FDD ephemeris and EMOS attitude data
  - Ephemeris data
    - Is received from the FDD in ephemeris data files
    - Ephemeris data arrives at the DAAC daily about eight to ten hours after the end of the UTC day
  - Attitude data
    - Ground-Based Attitude Determination (GBAD) data is used in the processing of attitude data
    - GBAD data is received from EMOS in "carry-out" files
    - GBAD carry-out files (PMCOGBAD) are used in conjunction with ephemeris data to prepare Aqua attitude data

Production Requests (Cont.)

- **Aqua DPREP (Cont.)**
  - **DPREP processing is granule-oriented**
    - **The processing interval selects data granules from the archive for DPREP to process**
    - **Then the granules get processed to completion**
    - **Ephemeris granules that are processed by Aqua DPREP consist of 24-hour segments**
    - **Attitude granules processed by Aqua DPREP consist of two-hour segments**

Production Requests (Cont.)

- **Aqua DPREP (Cont.)**
  - Aqua Ephemeris and Attitude PGEs are scheduled daily and run independently of one another
  - However, ephemeris is always run first on any given data segment because attitude processing depends on ephemeris data to complete its processing
  - For both Aqua DPREP PGEs, the size of the QA window is three records
    - A window of three data points contains the record undergoing QA analysis and the immediately preceding and following records
    - When performing QA analysis on records close to a granule boundary in the ephemeris and attitude data streams, the QA window extends into the preceding or following data granule as circumstances dictate
    - Consequently, Aqua DPREP needs nominal access to the granules that immediately precede and follow the "current" granule

Production Requests (Cont.)

- **Aqua DPREP: Ephemeris Processing**
  - **Includes reformatting the FDD ephemeris granules into Toolkit native format and HDF format**
  - **In addition, ephemeris metadata records are generated for the product granules**
  - **EcDpPrPm1FddEphemerisDPREP_PGE**
    - **Processes Aqua FDD ephemeris granules (i.e., PM1EPHD)**
    - **Output files/granules of EcDpPrPm1FddEphemerisDPREP_PGE provide satellite ephemeris data and are subsequently used in the processing of Aqua satellite attitude data**

Production Requests (Cont.)

- **Aqua DPREP: Ephemeris Processing (Cont.)**
  – **Because of the preprocessing that FDD performs on ephemeris data, DPREP performs minimal data quality checking**
    - **The data quality analyses include continuity, consistency, range, and data gap checking, which are intended mainly to catch transmission errors**
    - **DPREP does not perform limit (spike) checking or data repair on FDD ephemeris**
    - **Any failure of data quality analysis (e.g., data gap or consistency problem) detected in the FDD ephemeris triggers a request for a replacement granule**
    - **In such cases a subscription for the FDD ephemeris replacement granule is submitted to the Spatial Subscription Server**

Production Requests (Cont.)

- **Aqua DPREP: Ephemeris Processing (Cont.)**
  - **For the sake of timely FDD ephemeris processing, Aqua DPREP typically foregoes gap checking and continuity checking of the FDD ephemeris data timeline/data stream between the segment being processed and the one immediately following**
    - **Rather than wait 24 hours for the FDD ephemeris granule from the following segment to become available, DPREP omits gap checking and continuity checking at the following segment boundary (i.e., it reverts to Profile 3 processing)**
    - **Consequently, DPREP can detect such a data gap or ensure continuity only when processing of the following segment completes (e.g., during the next day's run of EcDpPrPm1FddEphemerisDPREP_PGE)**
  - **Note that there is no need to forgo gap checking or continuity checking during FDD ephemeris *reprocessing* because all granules on the timeline should be available in the archive**

Production Requests (Cont.)

- **Aqua DPREP: Ephemeris Processing (Cont.)**
  - **EcDpPrPm1FddEphemerisDPREP_PGE**
    - **Reads in the “current” FDD PM1EPHD ephemeris granule, the immediately preceding ephemeris granule, and (if available) the immediately following ephemeris granule**
    - **In addition, it reads in a previous PM1EPHND preprocessed Aqua platform ephemeris granule in native format**
    - **The outputs of the process are preprocessed Aqua platform ephemeris granules in native format (PM1EPHND) and HDF format (PM1EPHHD)**
    - **A metadata file is produced for each output data file**

Production Requests (Cont.)

- **Aqua DPREP: Attitude Processing**
  - **EcDpPrPm1AttitudeDPREP_PGE**
    - **Inputs are from ESDTs PMCOGBAD (EMOS-supplied Ground-Based Attitude Determination carry-out files) and PM1EPHND (preprocessed Aqua platform ephemeris data in native format)**
    - **Outputs are PM1ATTHR (Aqua attitude data in HDF-EOS format) and PM1ATTNR (Aqua attitude data in native format)**
  - **The attitude DPREP stream executes (nominally) twelve times per day at two-hour intervals**
  - **PMCOGBAD data is in the form of two-hour granules (12 granules per day)**
  - **Each PM1EPHND (ephemeris) granule represents 24 hours of data (one granule per day)**

Production Requests (Cont.)

- **Aqua DPREP: Attitude Processing (Cont.)**
  - **The EMOS-supplied attitude consists of two data streams that are subject to QA analyses:**
    - **Attitude data stream**
    - **Guidance, Navigation, and Control (GN&C) Status Word 2 data stream**
  - **The GN&C Status Word 2 data stream contains the mode that the on-board attitude system was in when the attitude of the Aqua platform was recorded (at eight-second intervals)**

Production Requests (Cont.)

- **Aqua DPREP: Attitude Processing (Cont.)**
  - **Because the attitude data are preprocessed by EMOS prior to processing by DPREP, DPREP performs minimal data quality checking on the EMOS attitude data stream**
    - **Attitude data quality analyses include continuity, consistency, range, and data gap checking, which are intended mainly to catch transmission errors**
    - **DPREP does not perform limit (spike) checking or data repair**
    - **Any failure of data quality analysis (with the exception of long data gaps) results in entering a subscription for a replacement granule from EMOS**
    - **The presence of a long gap in the EMOS attitude data timeline does not cause DPREP to request replacement data from EMOS (the gap is flagged as a long gap and remains as such in the EMOS attitude data timeline)**

Production Requests (Cont.)

- **Aqua DPREP: Attitude Processing (Cont.)**
  – **In general EMOS attitude processing requires raw input from the preceding segment**
    - **In order to complete continuity checks between the preceding and current data segments on the Status Word 2 data stream that is imbedded within the EMOS-supplied attitude granules**
  – **Raw input is also required from the following data segment**
    - **In order to complete continuity checks between the current and following data segments on the attitude data stream as well as the Status Word 2 data stream**

Production Requests (Cont.)

- **Aqua DPREP: Attitude Processing (Cont.)**
  - **Because DPREP's data quality analysis includes checks for continuity across granule boundaries, EMOS attitude processing must lag at least two hours behind "real time"**
    - **Due to exceptional Aqua DPREP processing requirements, the lag is usually much longer**
    - **DPREP expects the EMOS-supplied attitude granule that follows the segment being processed to be available for continuity checking (hence the processing lag)**
  - **Interpretation of the attitude data depends on the value of the GN&C Status Word 2 contained in the carry-out file GBAD data**
    - **In general if the value of Status Word 2 represents "fine point" mode, the attitude is acceptable for Aqua science processing**
    - **Any other value indicates inadequate attitude accuracy for Aqua science data processing**

Production Requests (Cont.)

- **Aqua DPREP: Attitude Processing (Cont.)**
  - **EcDpPrPm1AttitudeDPREP_PGE**
    - **Reads in the previous, current, and next EMOS-supplied attitude granules (PMCOGBAD)**
    - **Reads in the ephemeris (PM1EPHND) granule for the same time period (and adjacent granule if near a granule boundary)**
    - **Reads in the attitude granule (PM1ATTNR) produced during the previous run**
    - **Output of the process is a native-format attitude file (PM1ATTNR) and an HDF-format attitude file (PM1ATTHR)**
    - **A metadata file is produced for each output data file**

Production Requests (Cont.)

- **Aqua DPREP: Aqua Data Repair**
  - **Because FDD and EMOS have preprocessed the ephemeris and attitude data streams, data repair has been deemed unnecessary and is not performed**

Production Requests (Cont.)

- **Aqua DPREP: Profiles**
  - **As mentioned with regard to Terra DPREP processing Aqua DPREP data requirements extend beyond the current segment**
  - **Data from the preceding and following segments are used in performing consistency checks on the ephemeris and attitude data streams when the data streams bridge segment boundaries**
    - **However, there is no guarantee that data from the preceding and following segments will always be available because adjacent segments may be in different granules**
    - **Consequently, four data processing profiles have been developed for each of the Aqua DPREP PGEs to accommodate the various permutations of data availability**
    - **The profiles were developed in accordance with the same principles as the corresponding Terra DPREP profiles**

Production Requests (Cont.)

- **Aura DPREP**
  - **Operationally, Aura DPREP is very similar to Aqua DPREP**
  - **Aura DPREP processes:**
    - **EcDpPrAuraEphemerisDPREP_PGE**
FDD Ephemeris Processing
    - **EcDpPrAuraAttitudeDPREP_PGE**
EMOS Attitude Processing
  - **Sources of information on the Aqua DPREP PGEs and how to run them:**
    - **500-EMD-003, *Aura Spacecraft Ephemeris and Attitude Data Preprocessing***
    - **611-EMD-001, *Mission Operation Procedures for the EMD Project*, Chapter 26**
    - **File installed on the science processor hosts (e.g., e0spg11, g0spg11, or l0spg11) in the /usr/ecs/*MODE*/CUSTOM/data/DPS directory; i.e., "HowToRunAuraDPREP"**

Production Requests (Cont.)

- **Aura DPREP (Cont.)**
  - Includes the processing of FDD ephemeris and EMOS attitude data
  - Ephemeris data
    - Is received from the FDD in ephemeris data files
    - Ephemeris data arrives at the DAAC daily about eight to ten hours after the end of the UTC day
  - Attitude data
    - Ground-Based Attitude Determination (GBAD) data is used in the processing of attitude data
    - GBAD data is received from EMOS in "carry-out" files
    - GBAD carry-out files (AUCOGBAD) are used in conjunction with ephemeris data to prepare Aura attitude data

Production Requests (Cont.)

- **Aura DPREP (Cont.)**
  - **DPREP processing is granule-oriented**
    - **The processing interval selects data granules from the archive for DPREP to process**
    - **Then the granules get processed to completion**
    - **Ephemeris granules that are processed by Aura DPREP consist of 24-hour segments**
    - **Attitude granules processed by Aura DPREP consist of two-hour segments**

Production Requests (Cont.)

- Aura DPREP (Cont.)
  - Aura Ephemeris and Attitude PGEs are scheduled daily and run independently of one another
  - However, ephemeris is always run first on any given data segment because attitude processing depends on ephemeris data to complete its processing
  - For both Aura DPREP PGEs, the size of the QA window is three records
    - A window of three data points contains the record undergoing QA analysis and the immediately preceding and following records
    - When performing QA analysis on records close to a granule boundary in the ephemeris and attitude data streams, the QA window extends into the preceding or following data granule as circumstances dictate
    - Consequently, Aura DPREP needs nominal access to the granules that immediately precede and follow the “current” granule

Production Requests (Cont.)

- **Aura DPREP: Ephemeris Processing**
  - **Includes reformatting the FDD ephemeris granules into Toolkit native format and HDF format**
  - **In addition, ephemeris metadata records are generated for the product granules**
  - **EcDpPrAuraEphemerisDPREP_PGE**
    - **Processes Aura FDD ephemeris granules (i.e., AUREPHMF)**
    - **Output files/granules of EcDpPrAuraEphemerisDPREP_PGE provide satellite ephemeris data and are subsequently used in the processing of Aura satellite attitude data**

Production Requests (Cont.)

- **Aura DPREP: Ephemeris Processing (Cont.)**
  - **Because of the preprocessing that FDD performs on ephemeris data, DPREP performs minimal data quality checking**
    - **The data quality analyses include continuity, consistency, range, and data gap checking, which are intended mainly to catch transmission errors**
    - **DPREP does not perform limit (spike) checking or data repair on FDD ephemeris**
    - **Any failure of data quality analysis (e.g., data gap or consistency problem) detected in the FDD ephemeris triggers a request for a replacement granule**
    - **In such cases a subscription for the FDD ephemeris replacement granule is submitted to the Spatial Subscription Server**

Production Requests (Cont.)

- **Aura DPREP: Ephemeris Processing (Cont.)**
  - **For the sake of timely FDD ephemeris processing, Aura DPREP typically foregoes gap checking and continuity checking of the FDD ephemeris data timeline/data stream between the segment being processed and the one immediately following**
    - **Rather than wait 24 hours for the FDD ephemeris granule from the following segment to become available, DPREP omits gap checking and continuity checking at the following segment boundary (i.e., it reverts to Profile 3 processing)**
    - **Consequently, DPREP can detect such a data gap or ensure continuity only when processing of the following segment completes (e.g., during the next day's run of EcDpPrAuraEphemerisDPREP_PGE)**
  - **Note that there is no need to forgo gap checking or continuity checking during FDD ephemeris *reprocessing* because all granules on the timeline should be available in the archive**

Production Requests (Cont.)

- **Aura DPREP: Ephemeris Processing (Cont.)**
  - **EcDpPrAuraEphemerisDPREP_PGE**
    - **Reads in the “current” FDD AUREPHMF ephemeris granule, the immediately preceding ephemeris granule, and (if available) the immediately following ephemeris granule**
    - **In addition, it reads in a previous AUREPHMN preprocessed Aura platform ephemeris granule in native format**
    - **The outputs of the process are preprocessed Aura platform ephemeris granules in native format (AUREPHMN) and HDF format (AUREPHMH)**
    - **A metadata file is produced for each output data file**

Production Requests (Cont.)

- **Aura DPREP: Attitude Processing**
  - EcDpPrAuraAttitudeDPREP_PGE
    - Inputs are from ESDTs AUCOGBAD (EMOS-supplied Ground-Based Attitude Determination carry-out files) and AUREPHMN (preprocessed Aura platform ephemeris data in native format)
    - Outputs are AURATTH (Aura attitude data in HDF-EOS format) and AURATTN (Aura attitude data in native format)
  - The attitude DPREP stream executes (nominally) twelve times per day at two-hour intervals
  - AUCOGBAD data is in the form of two-hour granules (12 granules per day)
  - Each AUREPHMN granule represents 24 hours of data (one granule per day)

Production Requests (Cont.)

- **Aura DPREP: Attitude Processing (Cont.)**
  - **The EMOS-supplied attitude consists of two data streams that are subject to QA analyses:**
    - **Attitude data stream**
    - **GN&C Status Word 2 data stream**

Production Requests (Cont.)

- **Aura DPREP: Attitude Processing (Cont.)**
  - Because the attitude data are preprocessed by EMOS prior to processing by DPREP, DPREP performs minimal data quality checking on the EMOS attitude data stream
    - As with ephemeris data processing, the attitude data quality analyses include continuity, consistency, range, and data gap checking, which are intended mainly to catch transmission errors
    - DPREP does not perform limit (spike) checking or data repair
    - Any failure of data quality analysis (with the exception of long data gaps) results in entering a subscription for a replacement granule from EMOS
    - The presence of a long gap in the EMOS attitude data timeline does not cause DPREP to request replacement data from EMOS
    - The gap is flagged as a long gap and remains as such in the EMOS attitude data timeline

Production Requests (Cont.)

- **Aura DPREP: Attitude Processing (Cont.)**
  - **In general EMOS attitude processing requires raw input from the preceding segment**
    - **In order to complete continuity checks between the preceding and current data segments on the Status Word 2 data stream that is imbedded within the EMOS-supplied attitude granules**
  - **Raw input is also required from the following data segment**
    - **In order to complete continuity checks between the current and following data segments on the attitude data stream as well as the Status Word 2 data stream**

Production Requests (Cont.)

- **Aura DPREP: Attitude Processing (Cont.)**
  - **Because DPREP's data quality analysis includes checks for continuity across granule boundaries, EMOS attitude processing must lag at least two hours behind "real time"**
    - **Due to exceptional Aura DPREP processing requirements, the lag is usually much longer**
    - **DPREP expects the EMOS-supplied attitude granule that follows the segment being processed to be available for continuity checking (hence the processing lag)**
  - **Interpretation of the attitude data depends on the value of the GN&C Status Word 2 contained in the carry-out file GBAD data**
    - **In general if the value of GN&C Status Word 2 represents "fine point," "attitude hold," or "earth point" mode, the attitude is acceptable for Aura science processing**
    - **Any other value indicates inadequate attitude accuracy for Aura science data processing**

Production Requests (Cont.)

- **Aura DPREP: Attitude Processing (Cont.)**
  - **EcDpPrAuraAttitudeDPREP_PGE**
    - **Reads in the previous, current, and next EMOS-supplied attitude granules (AUCOGBAD)**
    - **Reads in the ephemeris (AUREPHMN) granule for the same time period (and adjacent granule if near a granule boundary)**
    - **Reads in the attitude granule (AURATTN) produced during the previous run**
    - **Output of the process is a native-format attitude file (AURATTN) and an HDF-format attitude file (AURATTH)**
    - **A metadata file is produced for each output data file**

Production Requests (Cont.)

- **Aura DPREP: Aura Data Repair**
  - **Because FDD and EMOS have preprocessed the ephemeris and attitude data streams, data repair has been deemed unnecessary and is not performed**

Production Requests (Cont.)

- **Aura DPREP: Profiles**
  - As mentioned with regard to Terra and Aqua DPREP processing Aura DPREP data requirements extend beyond the current segment
  - Data from the preceding and following segments are used in performing consistency checks on the ephemeris and attitude data streams when the data streams bridge segment boundaries
    - However, there is no guarantee that data from the preceding and following segments will always be available because adjacent segments may be in different granules
    - Consequently, four data processing profiles have been developed for each of the Aura DPREP PGEs to accommodate the various permutations of data availability
    - The Aura DPREP profiles were developed in accordance with the same principles as the corresponding Terra DPREP profiles

Production Requests (Cont.)

- **ASTER Expedited Data Processing**
  - The system supports the ingest of ASTER expedited granules (EDSs) from the EOS Data and Operations System (EDOS) and their immediate availability to selected ASTER Scientists
  - In addition, the system supports the production of ASTER Level 1A/1B expedited data products from the ingested expedited granules
  - The higher-level expedited data products are produced at the Land Processes (LP) DAAC

Production Requests (Cont.)

- Process for ASTER expedited data processing
  - ASTER EDS data granules are transferred to the LP DAAC from the Goddard Space Flight Center (GSFC) Earth Sciences (GES) DAAC via cross-DAAC transfer mechanism
  - ASTER Ground Data System (GDS) produces a daily Observation Schedule File (OSF)
    - Including the ASTER Observation Schedule and the ASTER One-Day Schedule
  - OSF file is sent daily to the LP DAAC (outside the context of ECS at the DAAC)
  - OSF is processed by the ASTER OSF Parser tool at LP DAAC (outside the ECS context)
    - Output is referred to as Parsed OSF (POSF) data files

Production Requests (Cont.)

- Process for ASTER expedited data processing (Cont.)
  - At LP DAAC the POSF files are inserted into the ECS system via the Science Investigator-Led Processing Systems (SIPS) interface mechanism
  - Production Planner deletes all successfully completed L1AE and L1BE DPRs from PDPS
  - Production Planner runs the ASTER Expedited Planning Script (i.e., "GetGranTimes.pl")
    - Script queries the SDSRV for the temporal metadata of any AST_EXP and AST_L1AE granules that were inserted to SDSRV since the last time it was run
    - Output is a set of candidate DPR start/stop times

Production Requests (Cont.)

- **Process for ASTER expedited data processing (Cont.)**
  - **Production Planner generates a set of Production Requests (PRs) for both the L1AE PGE and the L1BE PGE using the DPR start/stop times identified using the GetGranTimes.pl script**
    - **PGEs are configured to run with Collection Time (rather than Insertion Time) production rules**
  - **Production Planner generates PRs (Cont.)**
    - **If the matching AST_POSF granule has not been inserted, a L1AE PR is still generated with a subscription for AST_POSF**
    - **L1BE PGEs being planned are not those corresponding to the L1AE PGEs being planned; L1AE PGEs must be finished and their DPRs deleted from PDPS prior to planning the corresponding L1BE PGEs**

Production Requests (Cont.)

- Process for ASTER expedited data processing (Cont.)
  - Production Planner creates a new production plan, merging the newly created L1AE and L1BE PRs with any PRs from the currently active plan
  - Production Planner activates the new plan for execution of L1AE and L1BE DPRs
  - Subsequent processing, archiving and access activities are the same as for other data types

Production Requests (Cont.)

- **ASTER On-Demand Processing**
  - The system supports requests for on-demand high-level production
  - Requests can be divided into the following categories:
    - Pre-defined (standard) ASTER high-level products
    - High-level products with nonstandard input parameters
    - Digital Elevation Model (DEM) products
  - Requester selects the granule to be used and/or the additional non-standard parameters to generate the high-level product and submits a request for the system to create the high-level product
  - Granules produced by on-demand processing are not permanently archived

Production Requests (Cont.)

- **ASTER On-Demand Processing:**
  Standard or Non-Standard On-Demand Processing
  – Supports requests for processing of ASTER L1A and L1B data to higher information levels
  – A request may require a sequence of algorithms to be run on the specified data
  – Requester uses the EOS Data Gateway (EDG) web client to submit ASTER On-Demand Product Requests via the ECS V0 Gateway
  – On-Demand Manager (EcPlOdMgr) in PLS manages on-demand orders received from the EDG
    - Determines the order type
    - Creates the PRs and DPRs necessary to fill the order
    - If all input data are available, the DPR(s) is (are) submitted to the Data Processing Subsystem

Production Requests (Cont.)

- **ASTER On-Demand Processing:**
  Standard or Non-Standard On-Demand Processing (Cont.)
  - Job Management (EcDpPrJobMgmt) gets activated on-demand DPRs from PLS
  - If the on-demand DPRs are waiting for data to arrive or if there are not enough processing resources, Job Management places them in the On-Demand Queue
  - When all input data are available and there are adequate processing resources, the on-demand DPRs are released to AutoSys for processing
  - Execution Management (EcDpPrEM) sends updates to MSS concerning the status of the on-demand jobs

Production Requests (Cont.)

- **ASTER On-Demand Processing: DEM Production**
  - Requests for ASTER on-demand DEM production are submitted to the system through the EDG and are forwarded to the ASTER DEM SIPS for processing
    - The ASTER DEM SIPS is located in close proximity to the LP DAAC
  - When the ASTER DEM SIPS has created the requested product, ECS ingests it and distributes the product to the requester

Production Requests (Cont.)

- **ASTER On-Demand Processing: DEM Production (Cont.)**
  - The ASTER scientist submits a request for ASTER on-demand DEM production to ECS through the EDG
  - Upon receipt of an on-demand order, ODPRM determines the order type
  - The ODPRM submits a subscription (to the subscription server) to be notified when the ASTER DEM product has been inserted in the archive
  - The ODPRM sends an e-mail message to the ASTER DEM Operator
  - The ASTER DEM Operator creates the DEM in accordance with the order information

Production Requests (Cont.)

- **ASTER On-Demand Processing: DEM Production (Cont.)**
  - When the requested product has been created, the ASTER DEM operator puts a PDR file on the PDR server
    - The PDR file identifies the location of the ASTER DEM data file and the corresponding metadata file
  - ECS ingests the DEM product using the polling-with-delivery-record interface
  - Upon successful insertion of the DEM product in the archive, the science data server triggers the appropriate "insert event" qualified with the universal reference (UR) of the inserted granule
  - The subscription server queries the subscription database to determine which subscriptions need to be activated (fired)

Production Requests (Cont.)

- **ASTER On-Demand Processing: DEM Production (Cont.)**
  - **The status of the on-demand request is updated to "Processing Complete"**
  - **ODPRM receives subscription notification and sends a "user acquire" request to the science data server to have the on-demand product sent to the requester**
  - **The DEM product is sent to the requester via the requested medium**
    - **The data may be sent via ftp or on a physical medium (e.g., tape)**

Production Requests (Cont.)

- Reprocessing
  - ”Reprocessing" is used to indicate the following two situations:
    - A PGE has been run in the past and produced a product and for some reason (e.g., a file has been corrupted/deleted) the PGE needs to be run again to reproduce the exact same outputs
    - A PGE that was run in the past over a given time period or set of data has been improved (software or static inputs) by the Instrument Team, who would like to have the modified PGE run on the data from the previously processed time period/data set
  - The second case is the more important
    - Leads to the requirement that the system support a workload for reprocessing that is equal in size to the current or first-time processing load

Production Requests (Cont.)

- Reprocessing (Cont.)
  – PGE changes leading to reprocessing may occur for any of the following reasons:
    - Correct an error in the software
    - Improve the algorithm on the basis of better understanding of the instrument or physical phenomena
    - Use updated static files (e.g., calibration data)
    - Use new or different ancillary data files
    - Use different production rules
    - Use changed/additional run-time parameters
    - Use improved input resulting from changes to an upstream PGE

Production Requests (Cont.)

- **Reprocessing:**
  Operations Concepts
  - A PGE that is being used for reprocessing represents a new version of a PGE
  - A reprocessing PGE may specify new static or dynamic inputs rather than new PGE software
  - Output granules from reprocessing PGEs are represented under the existing ESDT used for the previous generation or version of the data granules
  - The distinction between generations of the data can be made through the ReprocessingActual attribute that is associated with each granule
    - Other mechanisms for distinguishing reprocessing levels include the ProductionDateTime and PGEVersion attributes

Production Requests (Cont.)

- **Reprocessing: Operations Concepts (Cont.)**
  - DPRs specifying the exact same PGE may be scheduled for both reprocessing and routine processing in the same production plan
    - For example, first-time processing may have been carried out for the time period from Day 1 to Day N, at which point an improved PGE is introduced
    - The improved PGE is used for processing from Day N+1 forward, and is also used concurrently in reprocessing from Day 1 to Day N
  - It is expected that in a typical planning session the Production Planner will plan time ranges for reprocessing that are comparable in extent with the routine processing
    - For example, the Production Planner may plan both routine processing and reprocessing for data acquired during 24-hour periods

Production Requests (Cont.)

- **Reprocessing:**
  Operations Concepts (Cont.)
  – **DPRs created via planning are organized and managed in separate queues by processing type**
    - **A fixed number of processing "slots" are reserved for on-demand processing, routine processing, and reprocessing**
    - **DPRs are dispatched for processing from either queue against the appropriate slot when the slot becomes available, i.e., as a job of that slot's type finishes and the input data become available**
    - **Operations personnel can configure the number of slots of each type**
    - **Operations personnel can change the number of slots of each type (and as a result the total number of concurrent processing slots) at run time**
    - **There is no dynamic slot allocation (allocation is done by Operations personnel)**

Production Requests (Cont.)

- Regeneration
  - Produce replacements for previously generated granules that have been lost or corrupted due to failure in the archive
  - General Process:
    - Retrieve the Production History file (PH) for the lost granule to determine parameters for the generation of replacement granules
    - Create Production Requests for the generation of replacement granules
    - Create and activate a Production Plan that includes the Production Requests for the generation of replacement granules
    - Prepare (if applicable) a “PDPS Residual Granules List,” which identifies granules that either cannot or should not be regenerated at the DAAC
    - Some granules do need not be reproduced; e.g., if there is a more recent version of the product

Regenerating Granules (Cont.)

- **Considerations that apply to the regeneration of granules:**
  - **All outputs of the PGE [not just those equivalent to the lost granule(s)] are to be produced and archived**
  - **Various factors could make it impossible to reproduce granules identical to the originals**
    - **There is no guarantee that when a PGE is re-run it will use the same inputs as were used during the original execution of the PGE; consequently, the output may be different from the original granule(s)**
    - **Variability of Optional/Alternate inputs, Ad Hoc Reprocessing, Metadata Checks, Metadata Query and other production rules affects PGE output**

Logging in to System Hosts

- **Logging in to system hosts is accomplished from a UNIX command line prompt**
  - **It is an initial set of steps that is performed when accomplishing many other Production Planning and Processing tasks**
- **Procedure**
  - **Access the command shell**
  - **Set the DISPLAY environmental variable**
  - **Log in to the specified host using secure shell and the specified user ID**

Launching the Production Request Editor

- **Production Request Editor-Associated Applications**
  - **Production Request Editor**
  - **Subscription Manager**
  - **Sybase ASE Server for the PDPS database**

Launching the Production Request Editor (Cont.)

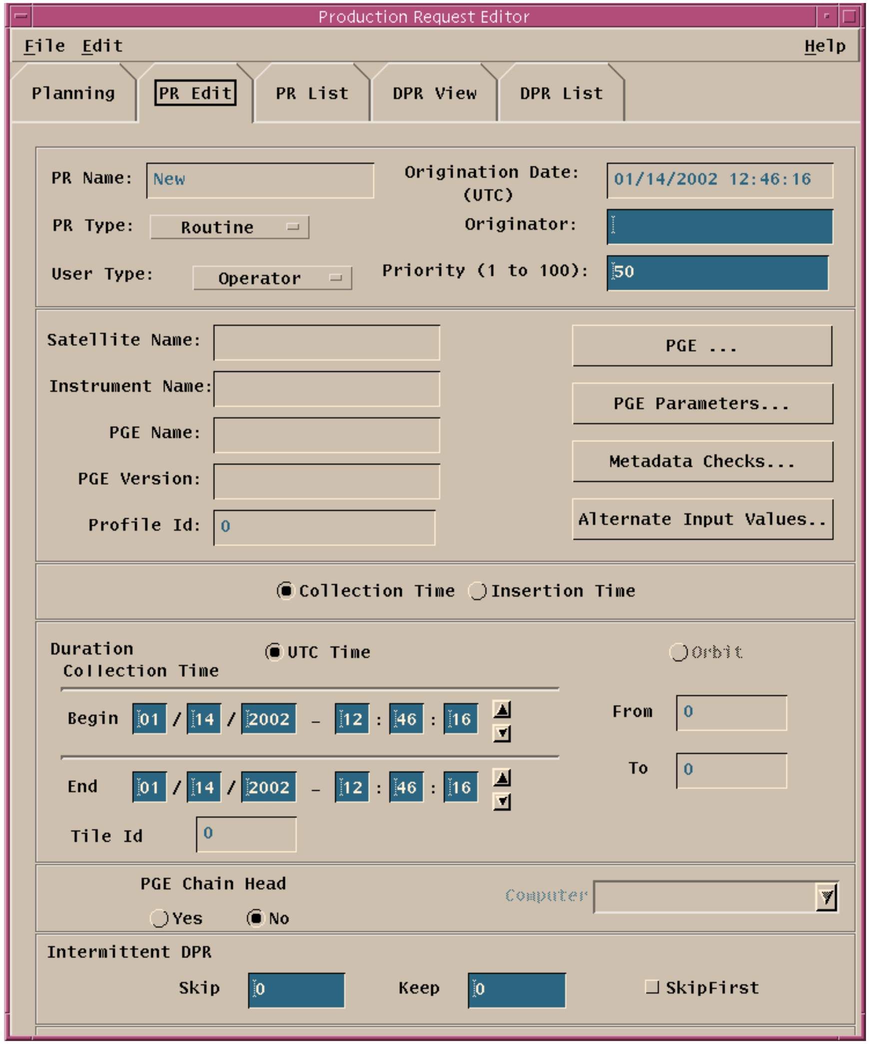
- **Procedure**
  - Access a terminal window logged in to the Planning/Management Workstation host
  - Set the ECS_HOME environmental variable if necessary
  - Change directory to the subdirectory (e.g., utilities) containing the production planning startup scripts
  - Start the Production Request Editor GUI in the appropriate mode

PR Editor Introductory GUI

Creating a New PR Using the Production Request Editor GUI

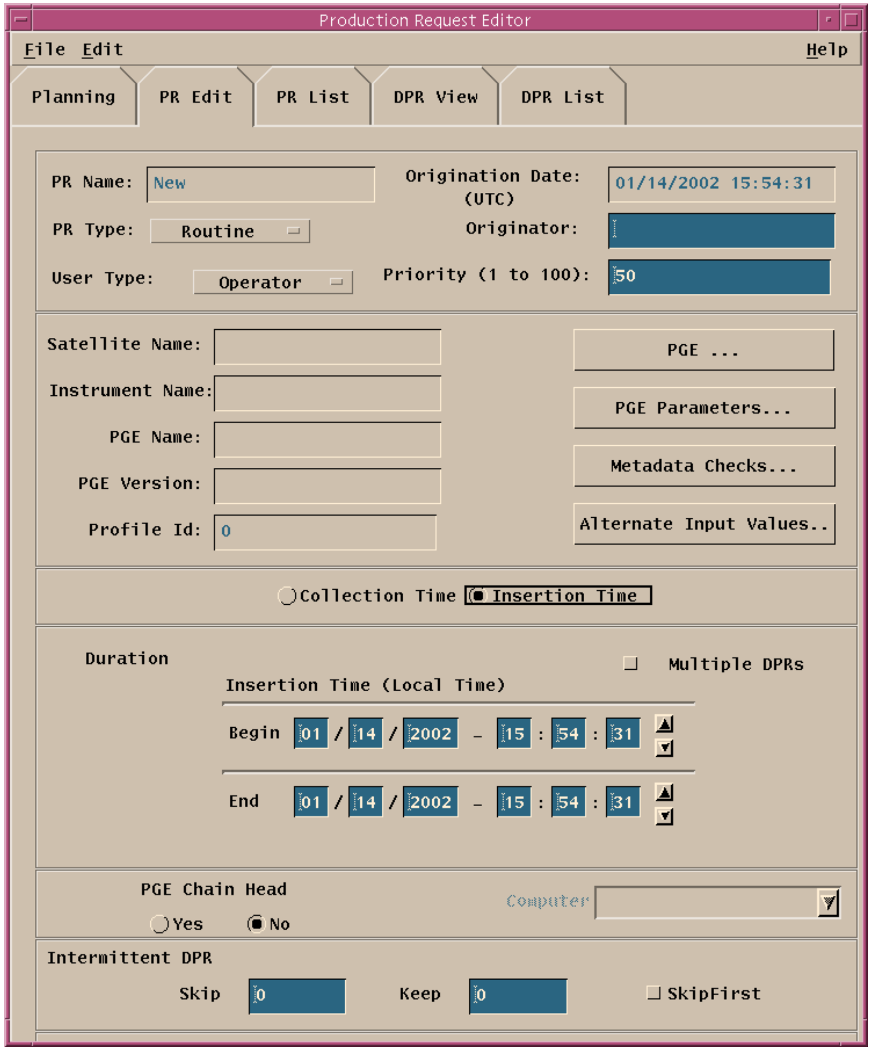
- **Procedure**
  - Select the PR Edit tab of the Production Request Editor GUI
  - Select the PGE to be included in the Production Request
  - If applicable, specify new metadata check value
  - If applicable, modify alternate input parameters
  - Specify the data requirements
    - Beginning and ending dates and times
    - Orbits (from/to)
    - Tile ID
  - Indicate whether the DPRs will be chain heads
  - If applicable, select the virtual computer for running the PGE
  - Save the Production Request

PR Edit GUI

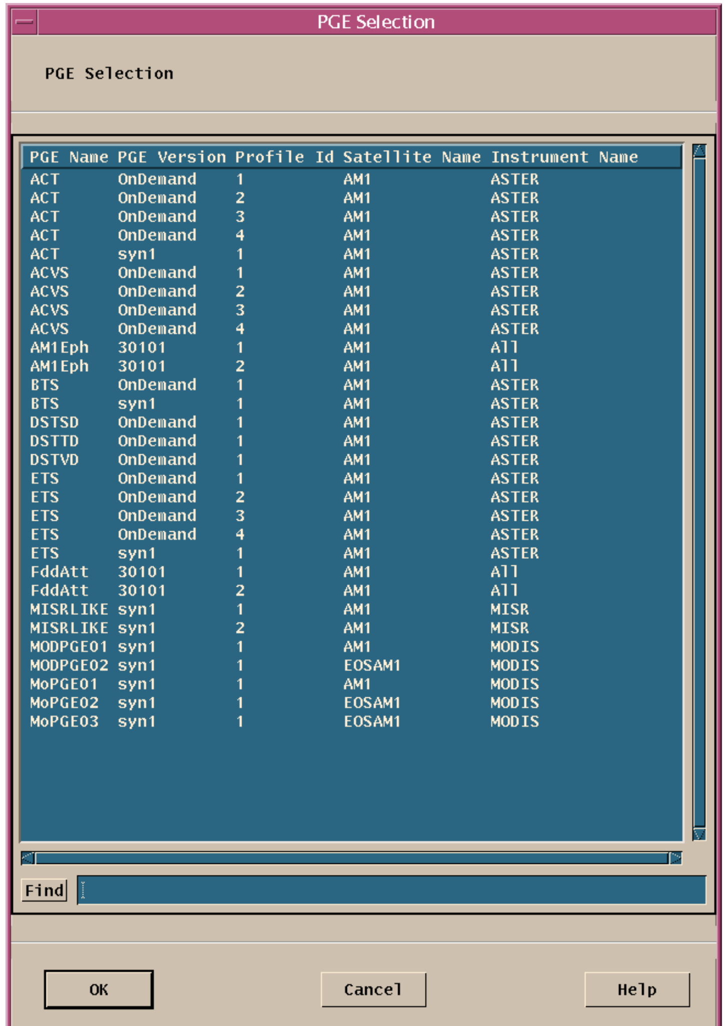


Production Request Editor
File Edit
Help
Planning
PR Edit
PR List
DPR View
DPR List
PR Name:
New
Origination Date: (UTC)
01/14/2002 12:46:16
PR Type:
Routine
Originator:
User Type:
Operator
Priority (1 to 100):
50
Satellite Name:
Instrument Name:
PGE Name:
PGE Version:
Profile Id:
0
PGE ...
PGE Parameters...
Metadata Checks...
Alternate Input Values..
Collection Time
Insertion Time
Duration Collection Time
UTC Time
Orbit
Begin
01 / 14 / 2002 - 12 : 46 : 16
End
01 / 14 / 2002 - 12 : 46 : 16
From
0
To
0
Tile Id
0
PGE Chain Head
Yes
No
Computer
Intermittent DPR
Skip
0
Keep
0
SkipFirst

PR Edit GUI with Multiple DPRs Toggle Button

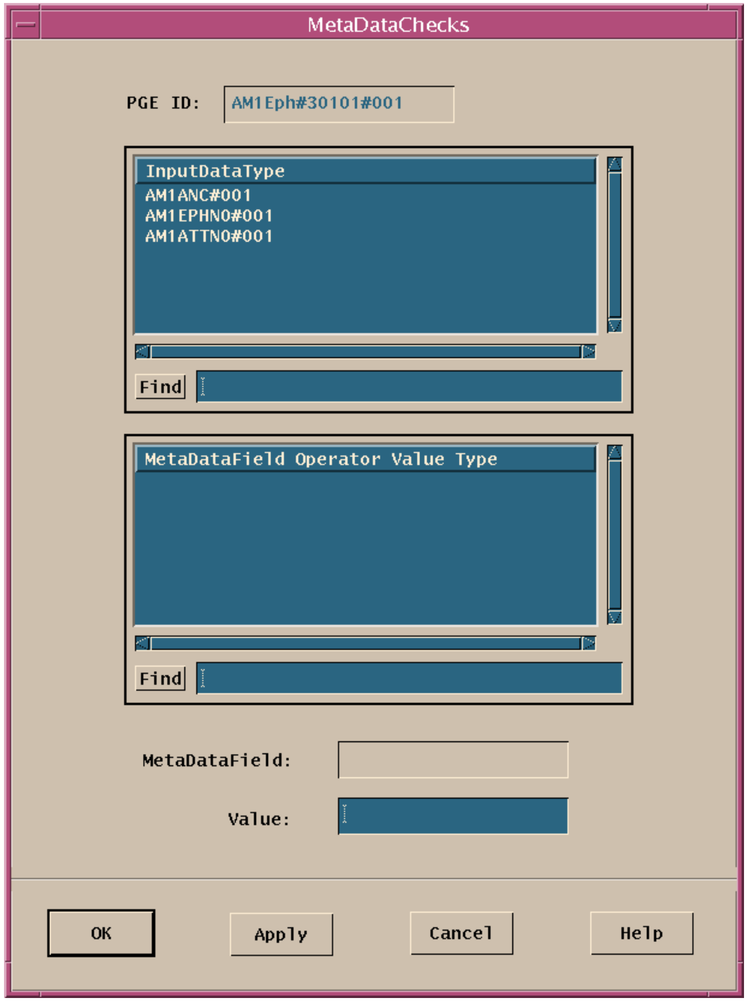


PGE Selection GUI

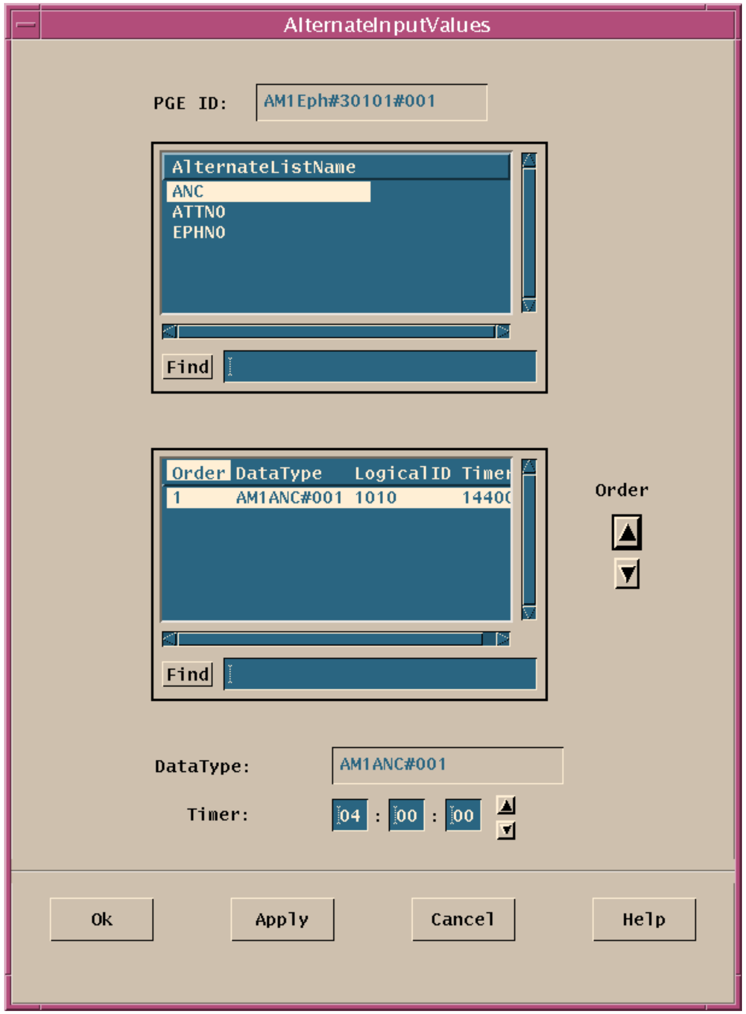


PGE Parameter Mappings GUI

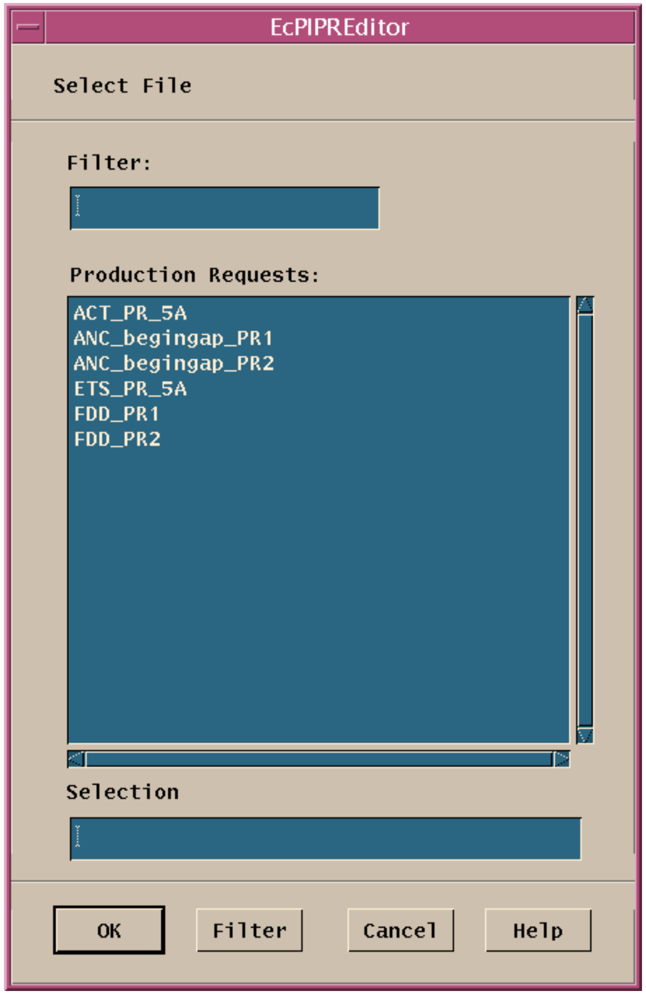
Metadata Checks GUI



Alternate Input Values GUI



PRE File Selection Window (“Open” or “Save As”)



Production Request Explosion into DPRs Dialogue Box

Creating New PRs Using the Production Request Generator

- **Procedure**
  - Log in to the appropriate host
  - Prepare an input file specifying the PgeId and GEOId values to be used in creating the production requests
  - Start the PR Generator to create new production requests
  - Check the PR Generator debug log to determine the results of running the PR Generator

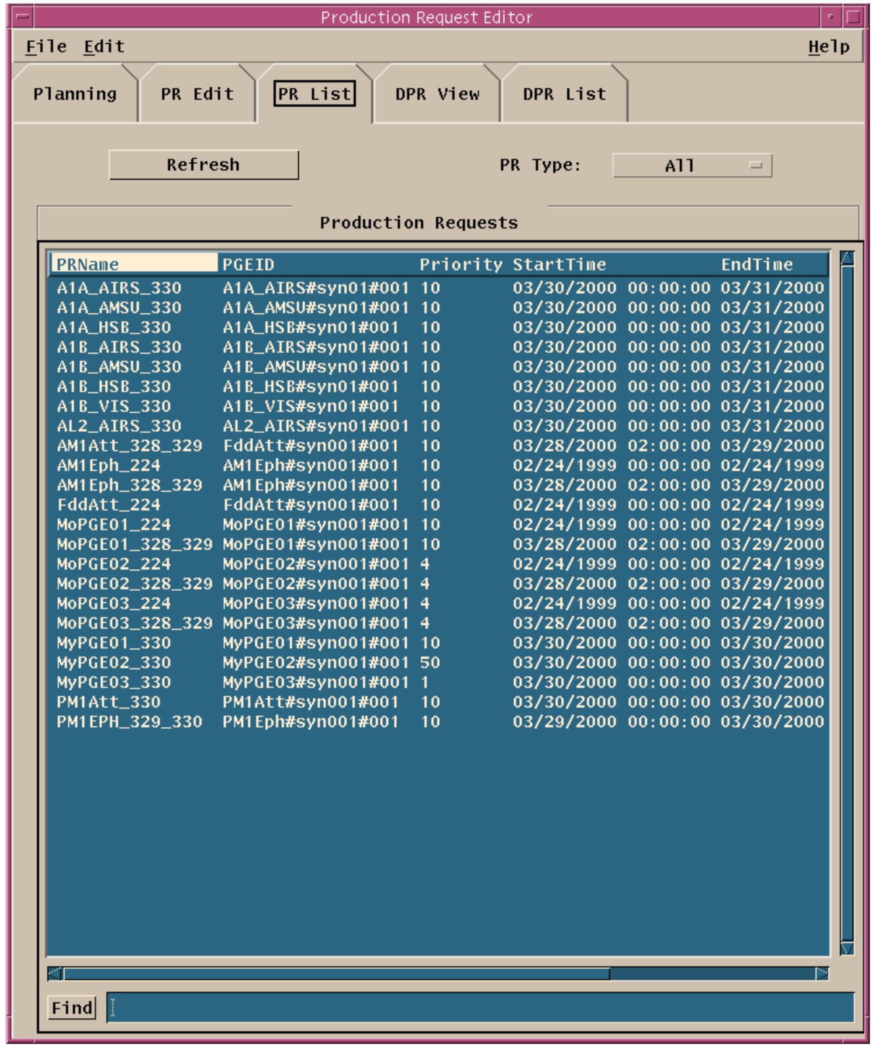
Editing/Modifying a PR

- **Procedure**
  - Select the PR Edit tab of the Production Request Editor GUI
  - Open the PR to be edited/modified
  - Make the necessary edits/modifications
  - Save the PR

Deleting a PR

- **Procedure**
  - **Select the PR List tab of the Production Request Editor GUI**
  - **Select the PR to be deleted from those listed**
  - **Select Edit→Delete**
  - **Select OK**
  - **Select OK**

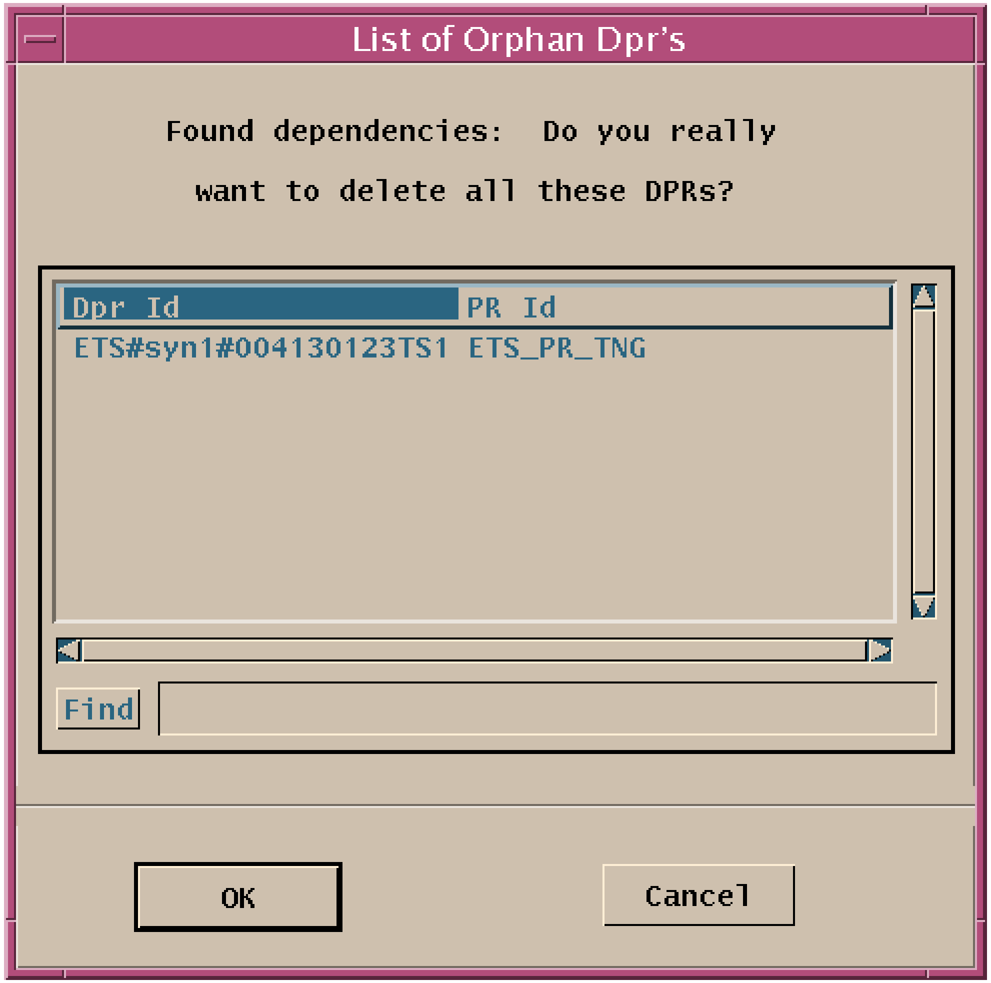
PR List GUI



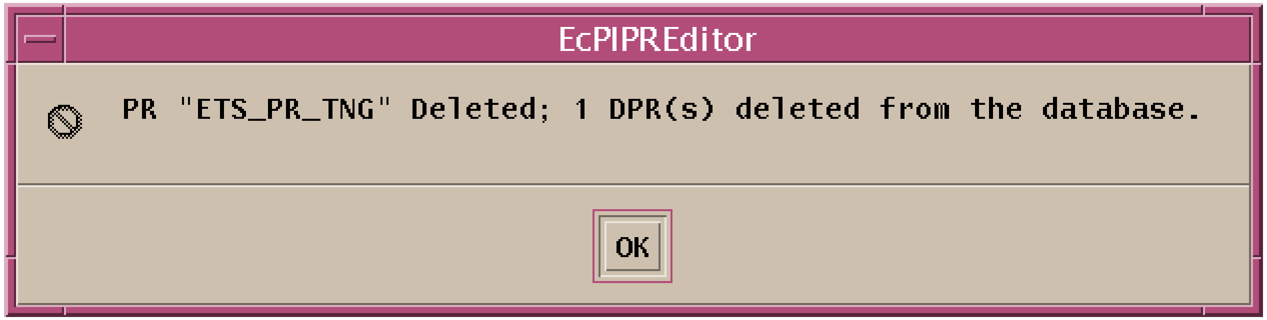
Production Request Editor
File Edit
Help
Planning
PR Edit
PR List
DPR View
DPR List
Refresh
PR Type:
All
Production Requests
PRName PGEID Priority StartTime EndTime
A1A_AIRS_330 A1A_AIRS#syn01#001 10 03/30/2000 00:00:00 03/31/2000
A1A_AMSU_330 A1A_AMSU#syn01#001 10 03/30/2000 00:00:00 03/31/2000
A1A_HSB_330 A1A_HSB#syn01#001 10 03/30/2000 00:00:00 03/31/2000
A1B_AIRS_330 A1B_AIRS#syn01#001 10 03/30/2000 00:00:00 03/31/2000
A1B_AMSU_330 A1B_AMSU#syn01#001 10 03/30/2000 00:00:00 03/31/2000
A1B_HSB_330 A1B_HSB#syn01#001 10 03/30/2000 00:00:00 03/31/2000
A1B_VIS_330 A1B_VIS#syn01#001 10 03/30/2000 00:00:00 03/31/2000
AL2_AIRS_330 AL2_AIRS#syn01#001 10 03/30/2000 00:00:00 03/31/2000
AM1Att_328_329 FddAtt#syn001#001 10 03/28/2000 02:00:00 03/29/2000
AM1Eph_224 AM1Eph#syn001#001 10 02/24/1999 00:00:00 02/24/1999
AM1Eph_328_329 AM1Eph#syn001#001 10 03/28/2000 02:00:00 03/29/2000
FddAtt_224 FddAtt#syn001#001 10 02/24/1999 00:00:00 02/24/1999
MoPGE01_224 MoPGE01#syn001#001 10 02/24/1999 00:00:00 02/24/1999
MoPGE01_328_329 MoPGE01#syn001#001 10 03/28/2000 02:00:00 03/29/2000
MoPGE02_224 MoPGE02#syn001#001 4 02/24/1999 00:00:00 02/24/1999
MoPGE02_328_329 MoPGE02#syn001#001 4 03/28/2000 02:00:00 03/29/2000
MoPGE03_224 MoPGE03#syn001#001 4 02/24/1999 00:00:00 02/24/1999
MoPGE03_328_329 MoPGE03#syn001#001 4 03/28/2000 02:00:00 03/29/2000
MyPGE01_330 MyPGE01#syn001#001 10 03/30/2000 00:00:00 03/30/2000
MyPGE02_330 MyPGE02#syn001#001 50 03/30/2000 00:00:00 03/30/2000
MyPGE03_330 MyPGE03#syn001#001 1 03/30/2000 00:00:00 03/30/2000
PM1Att_330 PM1Att#syn001#001 10 03/30/2000 00:00:00 03/30/2000
PM1EPH_329_330 PM1Eph#syn001#001 10 03/29/2000 00:00:00 03/30/2000
Find

Production Request Deletion Confirmation Dialogue Box

Production Request List of Orphan DPRs Dialogue



Production Request Deletion Completed Dialogue Box



Reviewing/Deleting Data Processing Requests (DPRs)

- Data Processing Requests (DPRs)
  - generated automatically by the PDPS
  - generated automatically from a PR (which specifies a PGE)
- DPR information is used by…
  - Data Processing Subsystem
  - AutoSys production scheduling software
- The Production Planner can review DPRs
- Operations personnel cannot edit DPR fields

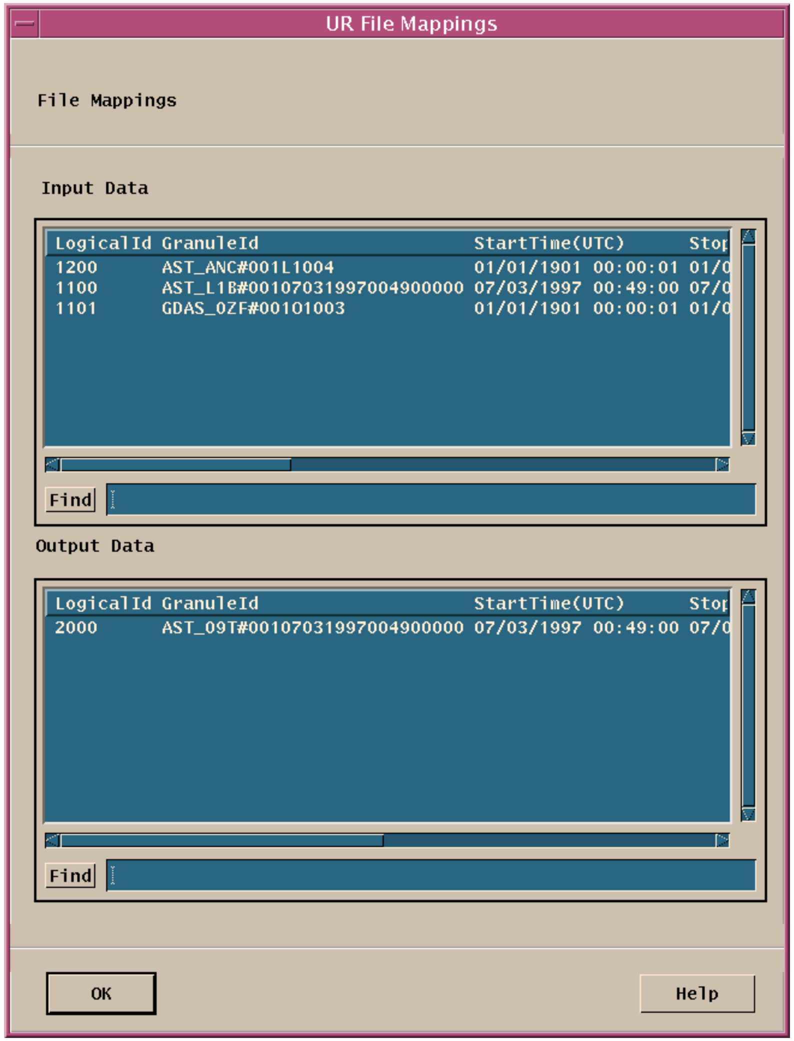
Reviewing/Deleting DPRs (Cont.)

- DPR List (Production Request Editor)
  - Each line of the Data Processing Request table represents a DPR, i.e., a job that will be run when all data and resource needs have been satisfied
  - For each DPR the table includes…
    - DPR identification
    - relevant PGE
    - name of the corresponding PR
    - data start date and time, etc.
  - may be filtered, so that only DPRs with certain characteristics are displayed

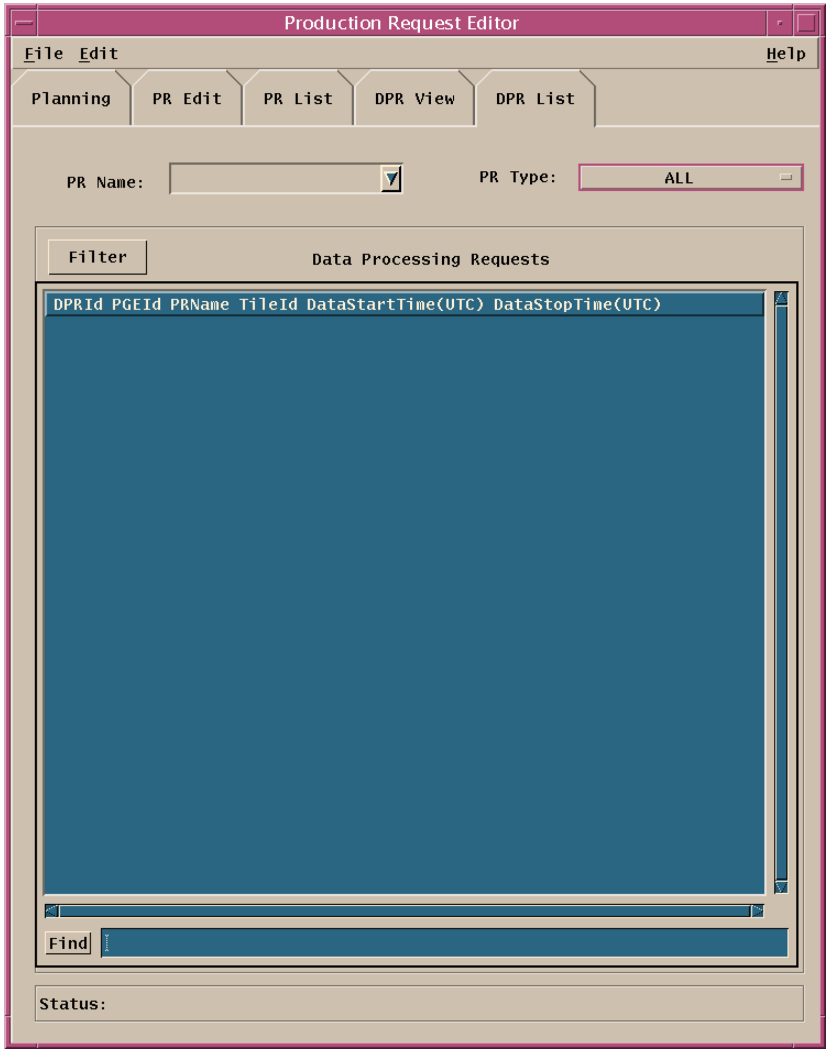
Reviewing/Deleting DPRs (Cont.)

- **Data concerning an individual DPR (Production Request Editor)**
  – **PGE parameters**
  – **UR File Mappings (PGE File Mappings)**
    - **input and output files for a particular DPR on the PGE File Mappings GUI**
    - **GUI displays one line of information for each file that may be used by or be produced by the PGE**

UR File Mappings GUI



DPR List GUI

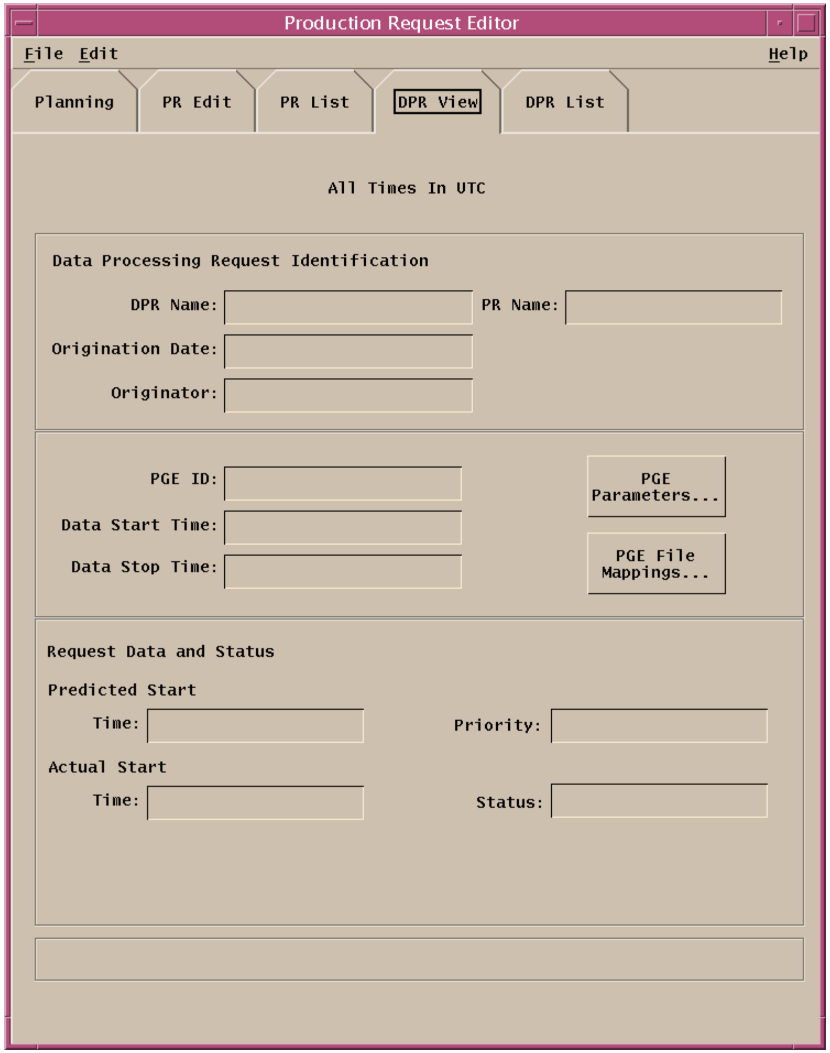


Production Request Editor
File Edit
Help
Planning
PR Edit
PR List
DPR View
DPR List
PR Name:
PR Type:
ALL
Filter
Data Processing Requests
DPRId PGEId PRName TileId DataStartTime(UTC) DataStopTime(UTC)
Find
Status:

Reviewing/Deleting DPRs (Cont.)

- **Procedure (Reviewing DPRs)**
  - Select the Data Processing Requests list by clicking on the DPR List tab
  - Select a Production Request from the list on the PR Name option button
  - Click on a DPR to be reviewed from the list of Data Processing Requests
  - Select File → Open from the pull-down menu
  - Click on the DPR View tab
  - Review the selected DPR
    - Click on the PGE Parameters… button to view the PGE parameters associated with the DPR
    - Click on the PGE File Mappings… button to view the UR file mappings (PGE input and output data)

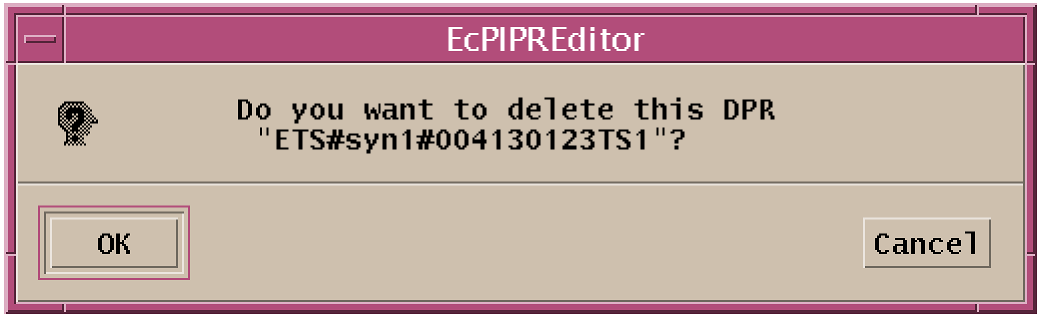
DPR View GUI


Production Request Editor
File Edit
Help
Planning
PR Edit
PR List
DPR View
DPR List
All Times In UTC
Data Processing Request Identification
DPR Name:
PR Name:
Origination Date:
Originator:
PGE ID:
Data Start Time:
Data Stop Time:
PGE Parameters...
PGE File Mappings...
Request Data and Status
Predicted Start
Time:
Priority:
Actual Start
Time:
Status:

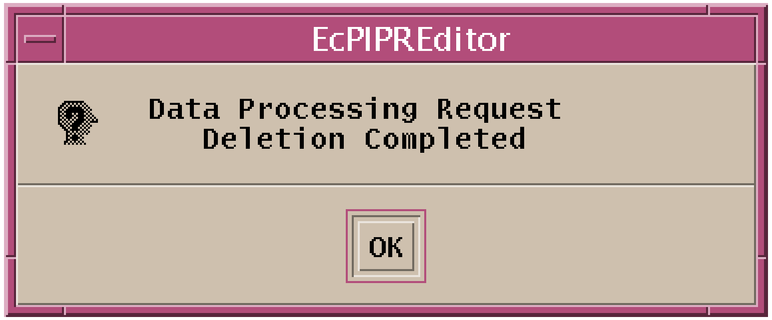
Reviewing/Deleting DPRs (Cont.)

- **Procedure (Deleting a DPR)**
  - Select DPR List tab on the Production Request Editor GUI
  - Select the appropriate Production Request from the list on the PR Name option button
  - Select the DPR to be deleted from the list of Data Processing Requests
  - Select Edit→Delete
  - Select OK
  - Select OK

DPR Deletion Confirmation Dialogue Box



DPR Deletion Completed Dialogue Box



Reviewing/Deleting DPRs (Cont.)

- Deleting or Aborting an On-Demand Processing Request
  - The Planning Subsystem detects changes in the status of ASTER on-demand processing requests
    - Whether the changes were made by the system or by operations personnel
  - PLS is not able to determine why operations personnel have canceled or aborted a request
  - Consequently, if it is necessary to cancel or abort an on-demand processing request, notify User Services personnel
    - So they can send an e-mail message to the requester explaining why the request was canceled or aborted

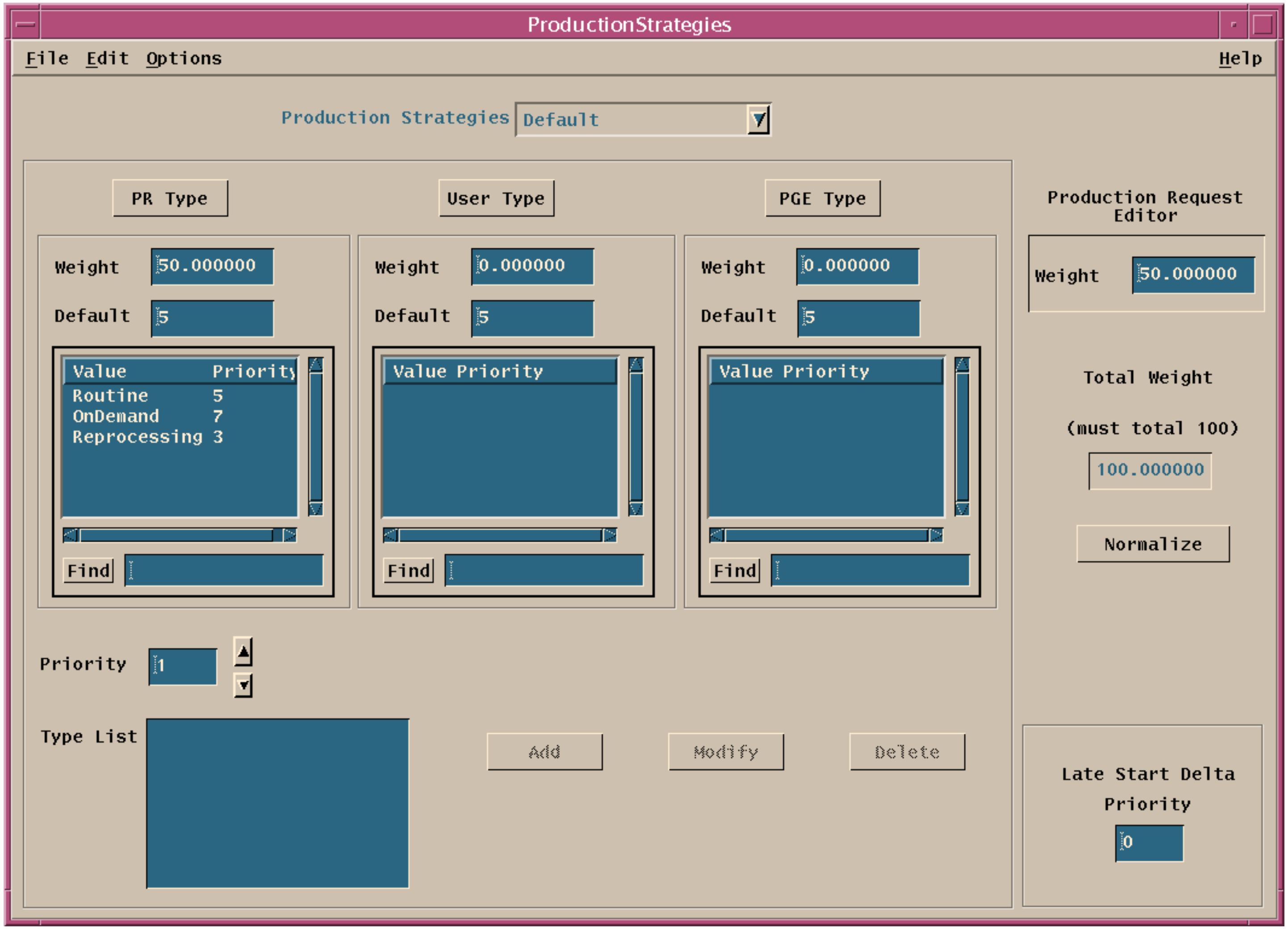
Launching Planning Workbench-Related GUIs

- **Launching the Production Strategies GUI**
  - **Software applications associated with the Production Strategies GUI**
    - **Production Strategies GUI**
    - **Sybase ASE Server for the PDPS database**
  - **Production Strategies are high-level sets of priorities that the Production Planner makes available to the Planning Workbench for determining the priorities and preferences in the processing of DPRs**
  - **Values included in the selected strategy are read by the Planning Workbench when prioritizing the DPRs in a production plan**

Launching Planning Workbench-Related GUIs (Cont.)

- **Launching the Production Strategies GUI: Procedure**
  - Access a terminal window logged in to the Planning/Management Workstation host
  - Set the ECS_HOME environmental variable if necessary
  - Change directory to the subdirectory (e.g., utilities) containing the production planning startup scripts
  - Start the Production Strategies GUI in the appropriate mode

Production Strategies GUI



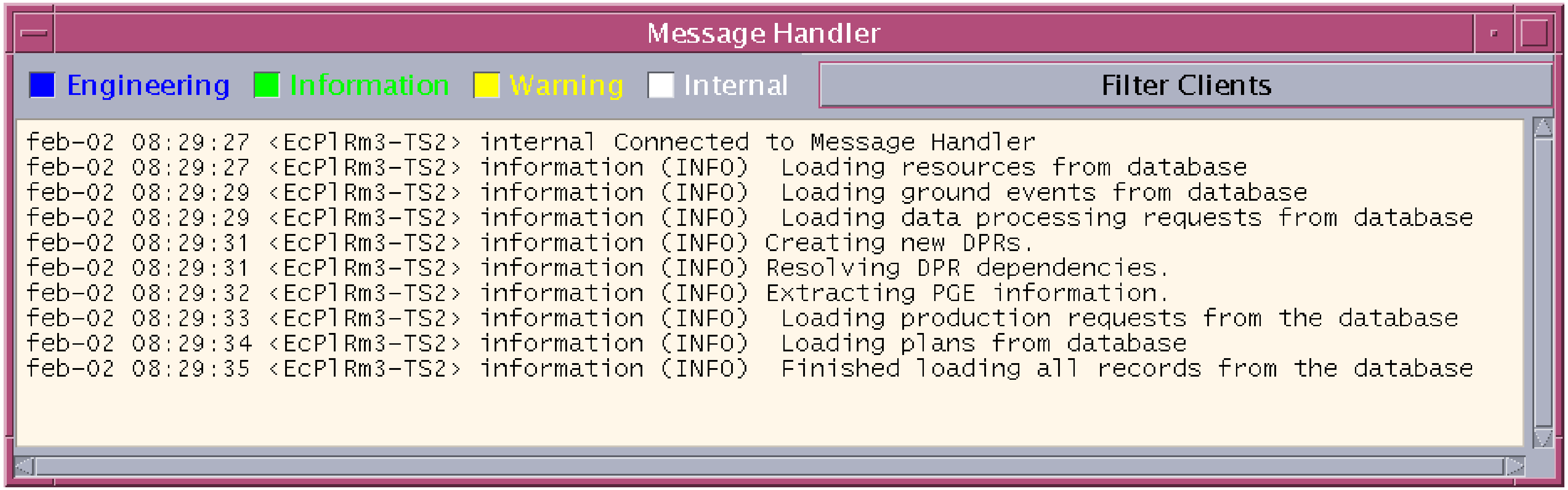
Launching Planning Workbench-Related GUIs (Cont.)

- **Planning Workbench-Related Applications**
  - System Name Server
  - Message Handler
  - Resource Model
  - Planning Workbench
  - Production Timeline
  - Production Strategies
  - Sybase ASE Server for the PDPS database

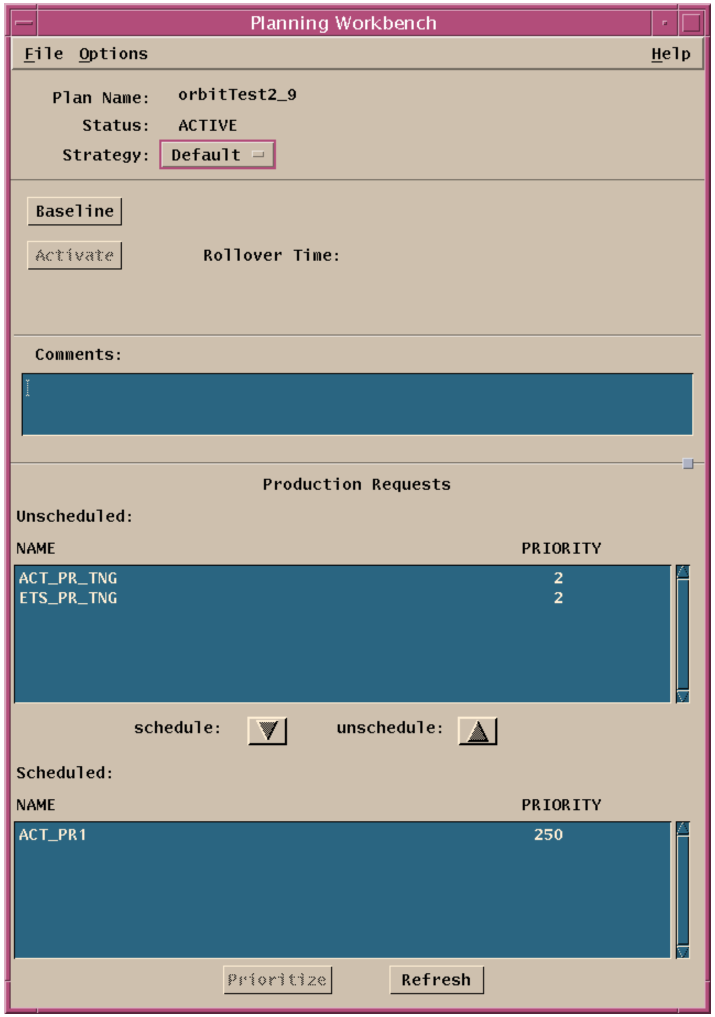
Launching Planning Workbench-Related GUIs (Cont.)

- **Launching Planning Workbench-Related GUIs: Procedure**
  - **Access a terminal window logged in to the Planning/Management Workstation host**
  - **Set the ECS_HOME environmental variable if necessary**
  - **Change directory to the subdirectory (e.g., utilities) containing the production planning startup scripts**
  - **Start the Planning Workbench GUI in the appropriate mode**
  - **NOTE: Normally, the EcPlAllStart script is used to start all processes. But if the number of DPRs in the PDPS database is very high (~4000), the EcPlSomeStart script is used to start the underlying processes. Then additional scripts are used to start the Planning Workbench GUI and the Timeline GUI.**

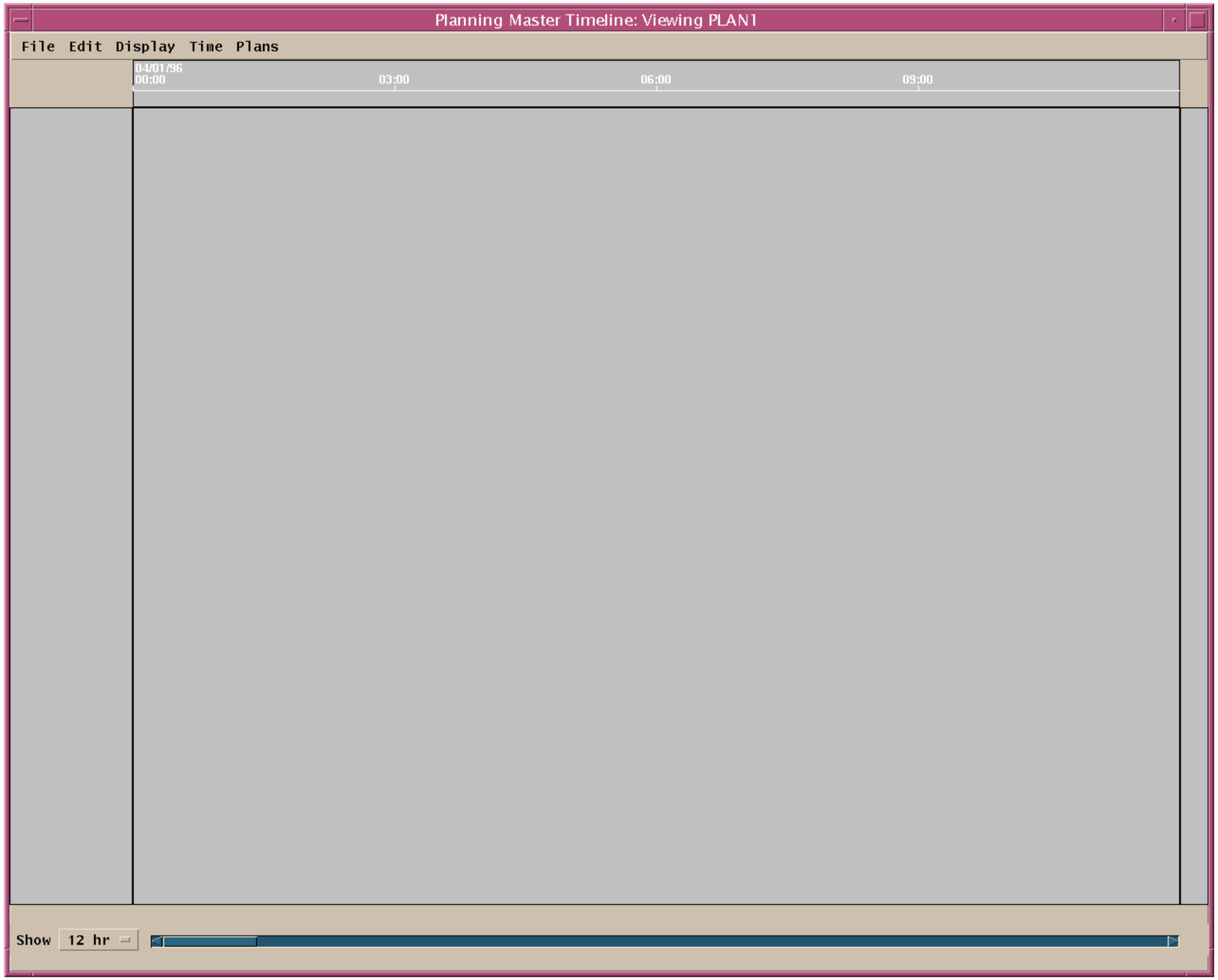
Message Handler GUI



Planning Workbench



Planning Timeline GUI



Creating a New Production Plan

- Planning process
  - Involves the Production Planner preparing monthly and weekly production plans as well as a daily production schedule from the most current weekly plan
    - Much of the planning occurs off line
    - However, the Production Planner uses the Planning Workbench GUI to create and activate the current or active plan
- Each DAAC has created its own policy and process for creating, reviewing, publishing and distributing production plans
  - Daily plans may be…
    - posted on a Worldwide Web page
    - distributed in hardcopy form
    - distributed via e-mail

Creating a New Production Plan (Cont.)

- Most important factor in production planning is meeting the customers' needs
  - Ultimate customers are the Science Computing Facilities that have provided science software to be run on the system on a routine basis
    - They expect timely generation and archiving of products using their science software
  - Factors to consider when creating production plans
    - Keeping up to date with routine processing of products using the SCFs' science software
    - Performing reprocessing of data using new versions of science software
    - Regenerating products (as needed) that have been damaged or lost from the archive
    - Taking into account the effects of ASTER on-demand processing, which circumvents the normal planning process but uses processing resources

Creating a New Production Plan (Cont.)

- Monthly plans
  - developed for the coming month and one or two months in advance
  - produced, reviewed, updated, published and distributed approximately two weeks before the beginning of the month
  - used to establish a baseline against which production targets can be measured
- Weekly plans
  - produced, reviewed, updated, published and distributed approximately five days before the beginning of the coming week
  - used to produce a baseline for comparison of planned vs. actual production results

Creating a New Production Plan (Cont.)

- Daily plan or schedule
  - produced each day for the next processing day
  - developed from the current weekly plan
    - adjusted to reflect the actual processing accomplished and the actual resources available at the time the daily schedule is generated
- Current (active) plan
  - activated as needed (one or more plans a day, depending on circumstances) using the Planning Workbench
  - implements the daily plan (or some portion of the daily plan)

Creating a New Production Plan (Cont.)

- During normal processing, when reasonably accurate predictions of the processing time for the PGEs are available, the processing schedule should result in a reasonably accurate prediction of when data products will be generated
- In abnormal situations (e.g., hardware or software failure), what is actually accomplished could depart significantly from the plan

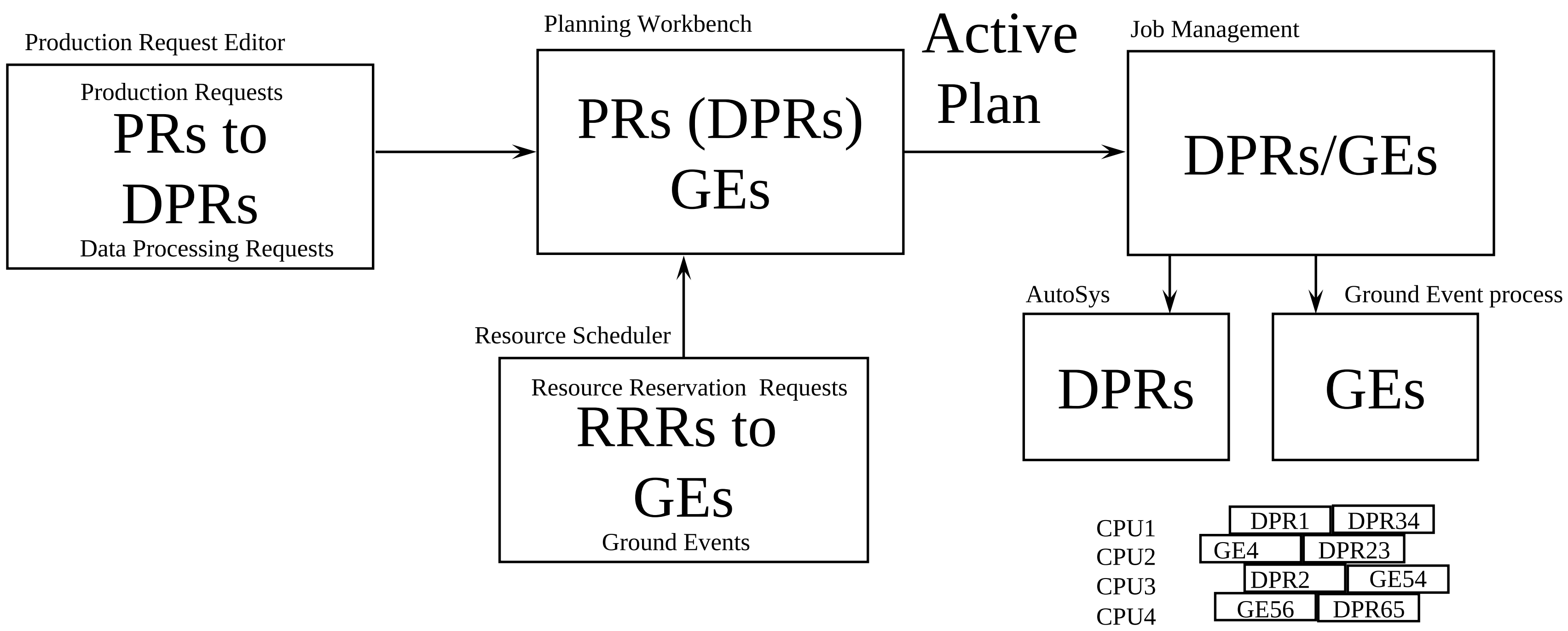
Creating a New Production Plan (Cont.)

- A daily plan may be divided into several segments to be activated separately
  - In such situations, the Production Planner is likely to develop a new current plan to continue processing or to change the processing priorities
  - The process is known as “replanning”
- Production Planner uses the Planning Workbench when creating a current (active) plan for production data processing at the DAAC
  - Planning Workbench provides the means by which the Production Planner selects and schedules specific PRs whose DPRs are to be run
  - Planning Master Timeline provides a forecast of the start and completion times of the jobs based upon experience in running the PGEs during SSI&T

Creating a New Production Plan (Cont.)

- When a plan is “activated” through the Planning Workbench, information in the plan is transferred to Job Management in the Data Processing Subsystem
  - Job Management creates and releases jobs into the Platinum AutoSys tool, where production processing is managed

Creating a New Production Plan: Planning Workbench



Production Strategy

- High-level set of priorities created by the Production Planner
- Made available for selection on the Planning Workbench
- Production Planner's choice of a strategy for a current production plan determines the priorities and preferences in the processing of associated DPRs

Production Strategy (Cont.)

- Production Strategies work on two levels:
  - Updating lists of DPR attributes so that each value an attribute can have is tied to a particular priority
  - Changing the weight that each attribute's priority is given
- In addition, weight is given to the priority selected by the user who entered the request

Production Strategy (Cont.)

- Total weights assigned to PR Type, User Type, PGE Type and Production Request Editor [Priority] must equal 100
- The values included in a strategy are read by the Planning Workbench to prioritize the DPRs in a plan
- Late Start Delta can be used to increase the priority of all jobs that have been waiting in the Production Queue for more than a day

Production Strategy: Calculating Priority for a DPR

DPR id:
MOD09:L2G_060199_14_1234
User Type:
Scientist
PGE:
MOD09:L2G
Routine request
Tile 14
Cluster id: 4
Priority (Production Request Editor): 7

PR TYPE	
Weight .45	
On Demand	**10**
Routine	6
Reprocessing	4
Default	5

PGE	
Weight .20	
MODIS 01	5
MOD09:L2G	**3**
MOD09:L3	6
MODIS 02	5
Default	5

USER TYPE	
Weight	.15
Operator	5
DAAC Manager	7
Researcher	10
Scientist	**8**
Default	5

Production Request Editor	
Weight	.20

The operator can also change the weights given to each list according to the DAACs current production priorities

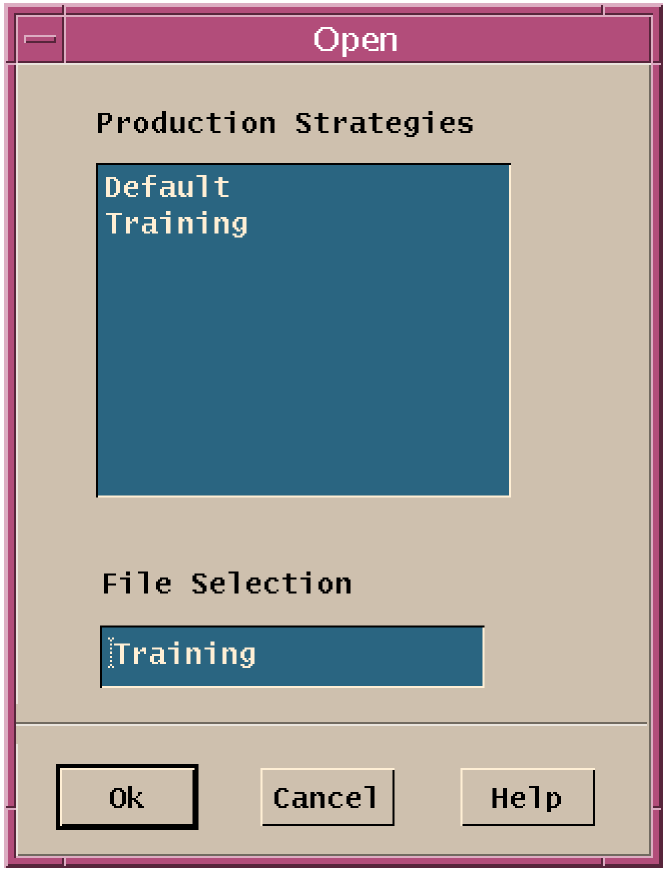
Priority = 6 * .45 +
3 * .20 +
8 * .15 +
7 * .20
= 5.9 = 6

The operator can change any weight or priority on any strategy list.

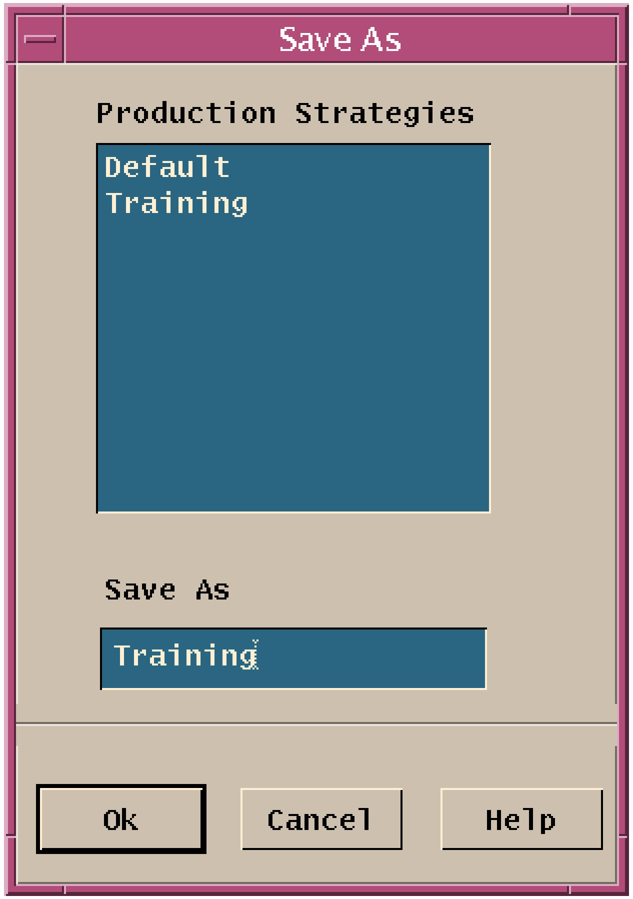
Defining a Production Strategy

- **Procedure**
  - **Select priorities for the values for each of the following three DPR attributes (as needed):**
    - **PR Type**
    - **User Type**
    - **PGE Type**
  - **Type weights for the preceding three DPR attributes (as needed)**
  - **Type a weight in the Production Request Editor field**
  - **Click on the Normalize button**
  - **Type delta priority for Late Start Delta (if needed)**
  - **Save the Production Strategy**

Production Strategies "Open" Window



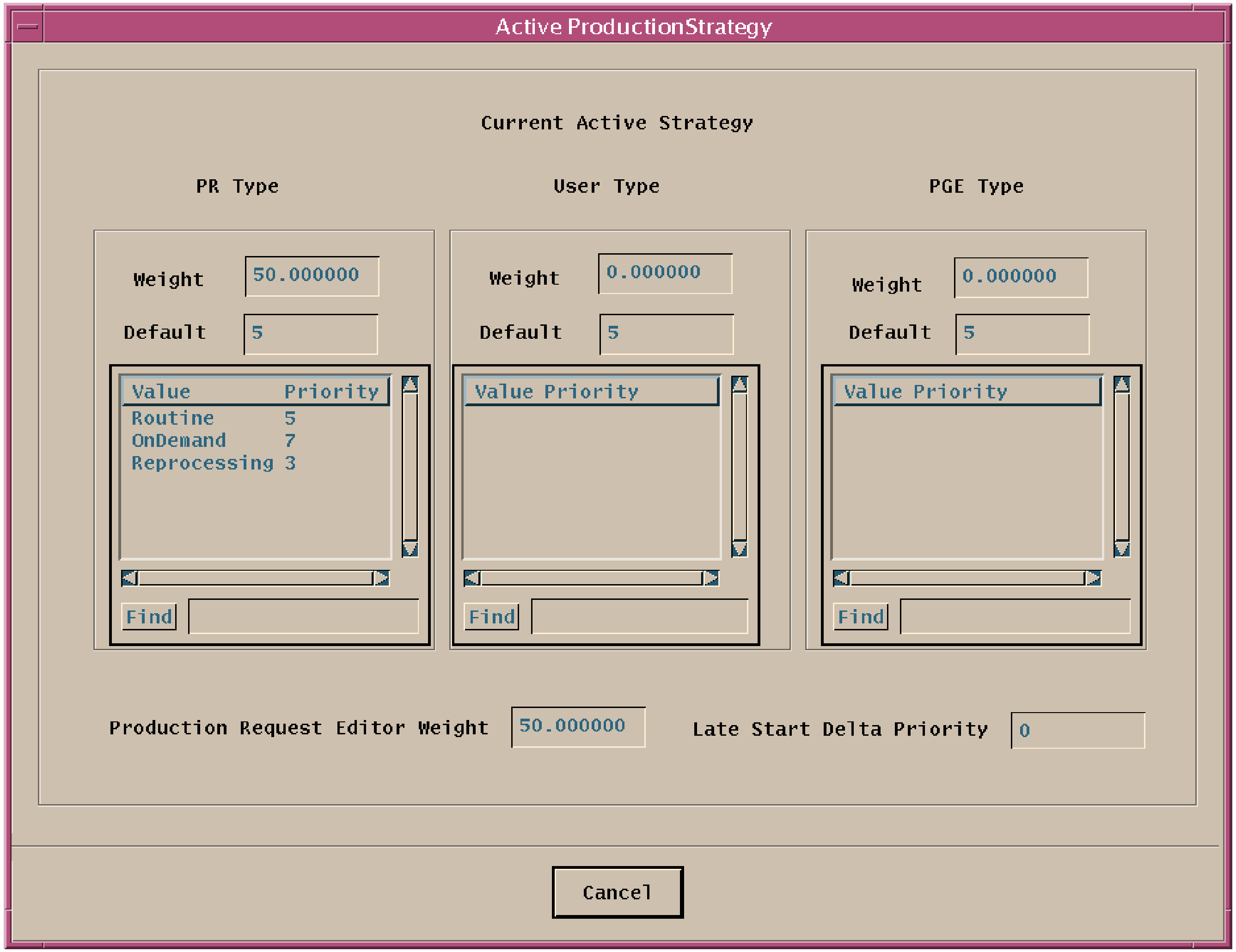
Production Strategies "Save As" Window



Reviewing the Current Active Strategy

- **Procedure**
  - Select Options → activeStrategy from the pull-down menu of the Production Strategies GUI
  - Review the data displayed in the Active Production Strategy window
    - PR Type values/weight
    - User Type data values/weight
    - PGE Type values/weight
    - Production Request Editor weight
    - Late Start Delta Priority value

Active Production Strategy Window



Deleting a Production Strategy

- **Procedure**
  - On the Production Strategies GUI click on the option button associated with the Production Strategies field, then highlight (in the option menu) the name of the production strategy to be deleted
    - Alternatively, it is possible to select File → Open from the pull-down menu, select the desired production strategy from the list on the Open window, and click on the Ok button to open the production strategy
  - Select Edit → Delete from the pull-down menu
  - Click on the OK button

Production Strategy Deletion Confirmation Dialogue Box

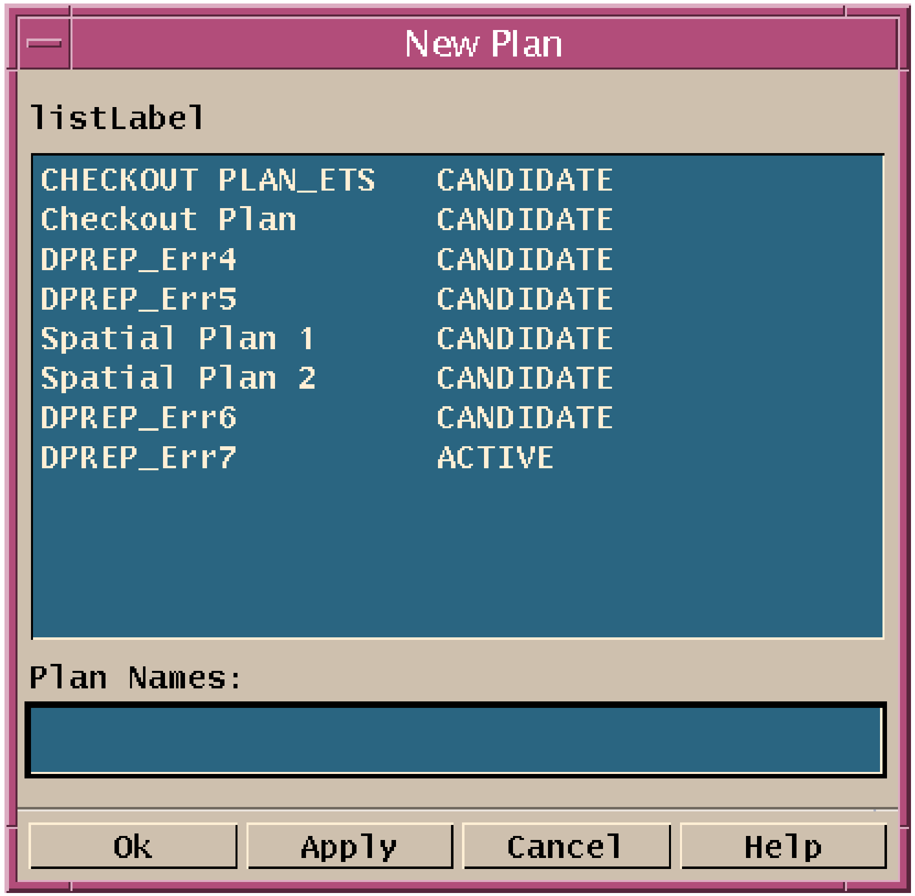
Creating a New Production Plan

- **Production Planner creates a plan for production data processing at the DAAC**
  - **selects specific PRs whose DPRs are to be run**
  - **selects PRs from two lists of PRs**
    - **list of available “Unscheduled” PRs**
    - **list of “Scheduled” PRs**
  - **uses arrow buttons to move PRs between lists until the “Scheduled” list contains the desired set of PRs that define the new plan**

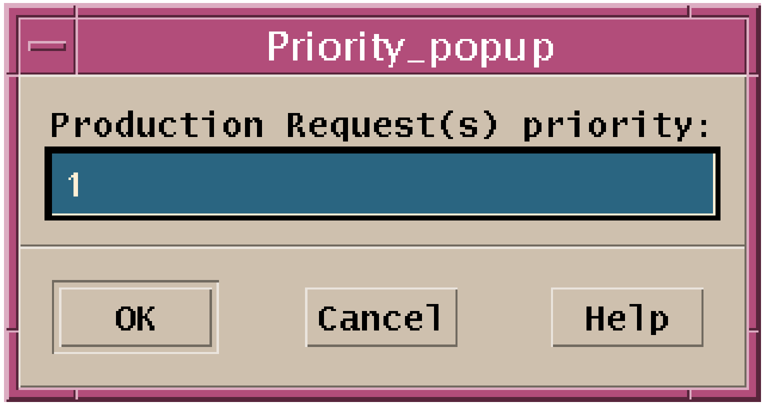
Creating a New Production Plan (Cont.)

- **Procedure**
  - Select applicable Production Strategy
  - Move PRs between the Unscheduled and Scheduled lists
  - Save the plan
  - Activate the plan if applicable
  - Save the plan as a baseline plan if applicable
  - To quit the Planning Workbench GUI: File → Exit
  - After quitting the Planning Workbench GUI, type the command to shut down the Message Handler, System Name Server, and Resource Model
  - Verify that the Message Handler, System Name Server, and Resource Model have shut down

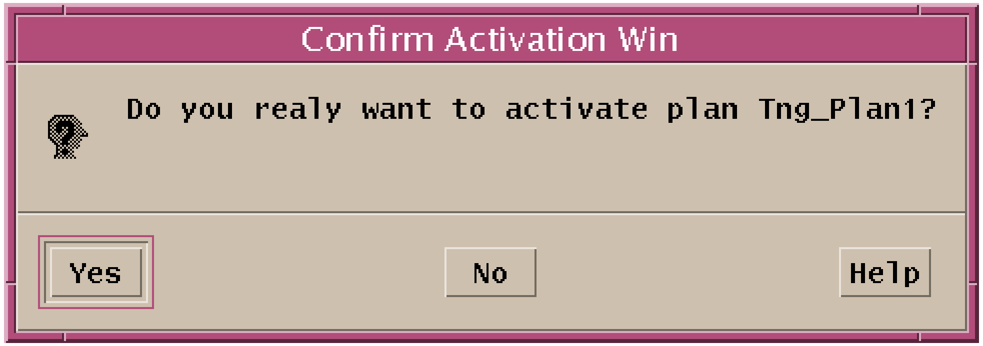
Planning Workbench “New Plan” GUI



Planning Workbench Priority Popup Window



Planning Workbench
Confirm Activation Dialogue Box



Creating a New Production Plan (Cont.)

- **Reactivation/Replanning**
  - **Assumes that there is a current plan**
  - **Current plan may or may not have DPRs associated with it**
    - **If there are no DPRs in a plan, it may be because all the DPRs that were in the plan have run to completion**
    - **Alternatively, there may never have been any DPRs associated with the plan because an empty plan was activated to cancel DPRs from a previously activated plan**
  - **Occurs in the following three types of situations:**
    - **No DPR in old plan but DPR(s) in new plan**
    - **DPR(s) in old plan but not in new plan**
    - **DPR(s) in both old plan and new plan**

Creating a New Production Plan (Cont.)

- Reactivation/Replanning (Cont.)
  - No DPR in Old Plan but DPR(s) in New Plan
    - Enter a new plan name (the plan is created using the Planning Workbench GUI)
    - Select the PRs to be included
    - Schedule the PRs to be activated
    - Save the new plan
    - Activate the new plan (new plan is activated and the new DPRs are ready to run)
    - Check on the state of the jobs in AutoSys (new DPRs should be in AutoSys and should begin to run)

Creating a New Production Plan (Cont.)

- **Reactivation/Replanning (Cont.)**
  - **DPR(s) in Old Plan but Not in New Plan**
    - **Verify that DPRs in the active plan are in the Job Management queue (current DPRs should be in the queue)**
    - **Create and save a new plan without PRs/DPRs to replace the old plan (with DPRs) - do not schedule any PRs to be activated**
    - **Activate the new plan**
    - **Verify the state of the new plan's DPRs (use the AutoSys JobScape GUI)**
    - **Verify the state of the old (existing) plan's DPRs (only jobs that are in the Job Management queue are cancelled; jobs that have already been released to AutoSys are not deleted during the replan - they continue processing to termination)**

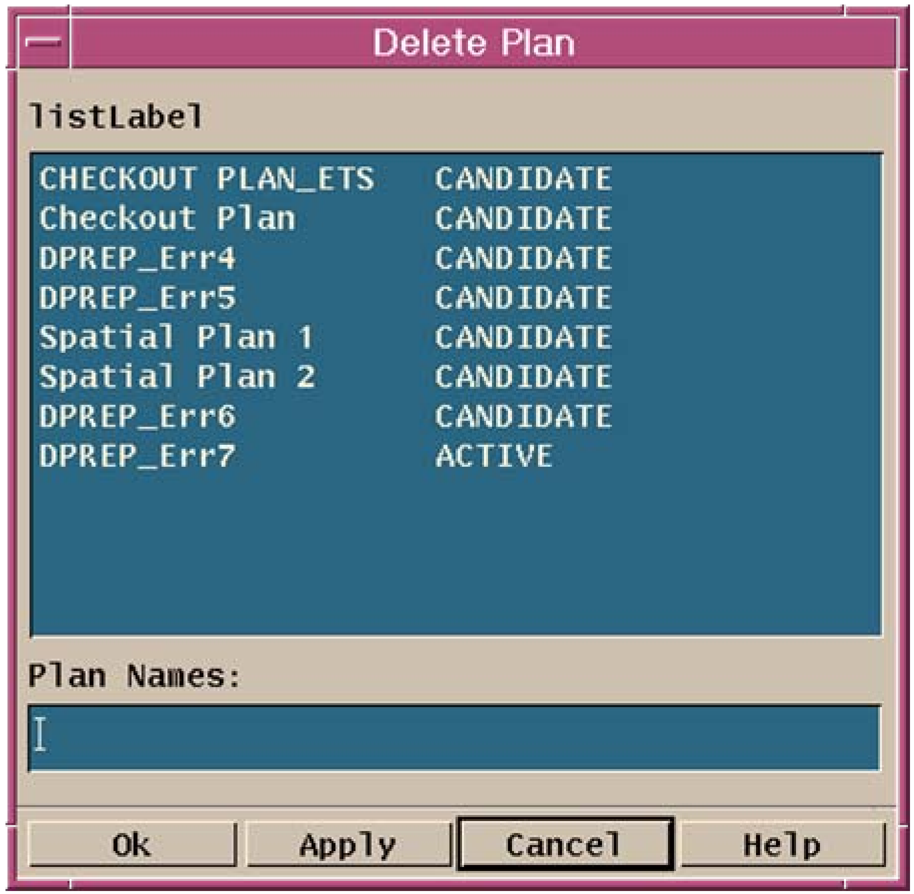
Creating a New Production Plan (Cont.)

- **Reactivation/Replanning (Cont.)**
  - **DPR(s) in Both Old Plan and New Plan**
    - **Verify that DPRs in the active plan are in the queue (current DPRs should be in the Job Management queue)**
    - **Enter a new plan name**
    - **Select the PRs to be included (include relevant PRs used in the old plan; i.e., those PRs with DPRs to be re-prioritized that are in the Job Management queue)**
    - **Schedule the PRs to be activated**
    - **Save the new plan**
    - **Activate plan**
    - **Verify the priorities of the plan's DPRs in the Job Management queue (new DPRs must be in the Job Management queue; old DPRs must be cancelled)**

Deleting a Production Plan

- **Procedure**
  - Select File → Delete on the Planning Workbench GUI
  - Select (highlight) the production plan to be deleted by clicking on the corresponding name in the list of plans
  - Click on the OK button

Planning Workbench Delete Plan Window



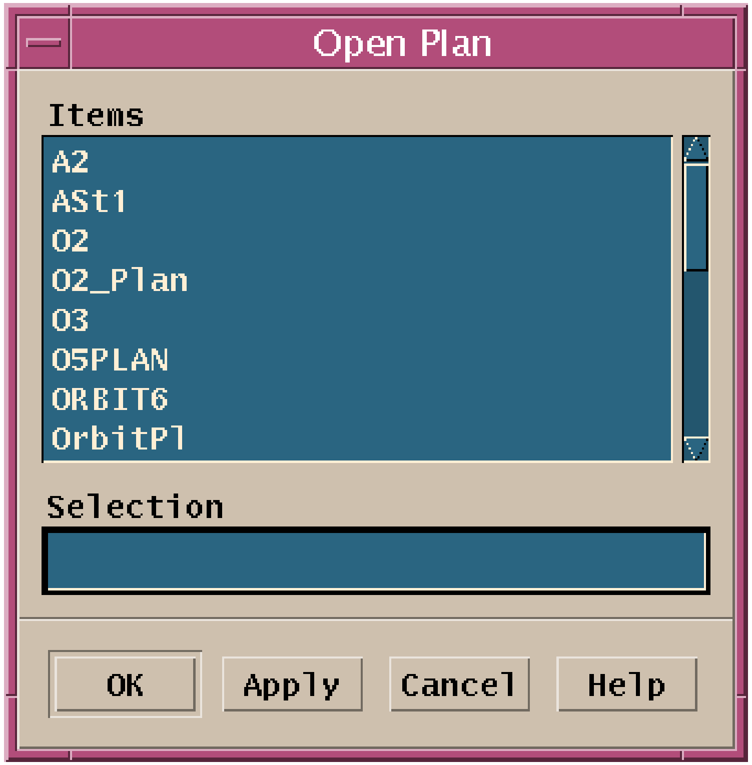
Reviewing a Plan Timeline

- Production Plan Timeline
  - graphic, timeline-oriented depiction of a production plan
  - displays a set of processing equipment, arranged along the left side of the GUI
  - displays some period of time across the top edge of the GUI
  - bars on the timeline represent either...
    - execution of DPRs on processing equipment over a period of time
    - resource reservations for non-production-related purposes (also called “ground events”)
  - ground events include such activities as testing, preventive maintenance, or system upgrades

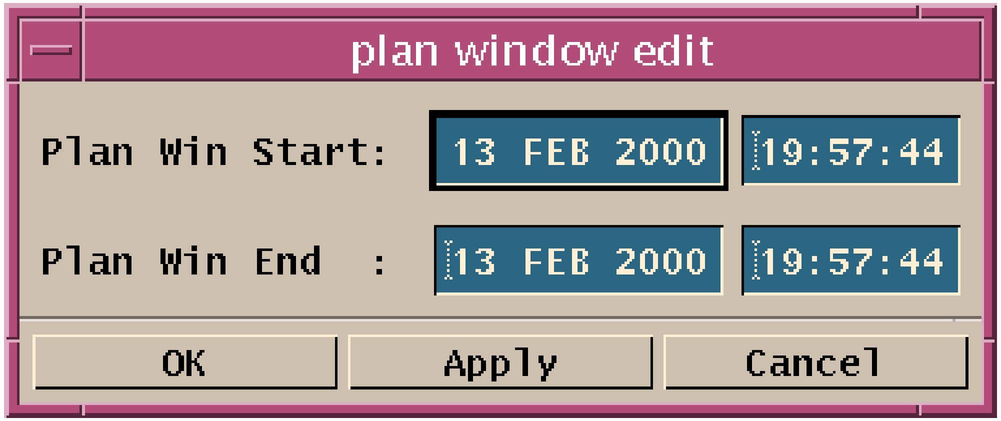
Reviewing a Plan Timeline (Cont.)

- Procedure
  - Adjust the Production Planning Timeline window size and the view of the timeline as necessary
  - Select a different plan to be viewed if necessary
  - Adjust the time scale (start and end dates and times) as necessary
  - Adjust the time span if desired
  - Adjust the resources to be displayed on the timeline as necessary
  - Adjust timeline color coding if desired/necessary

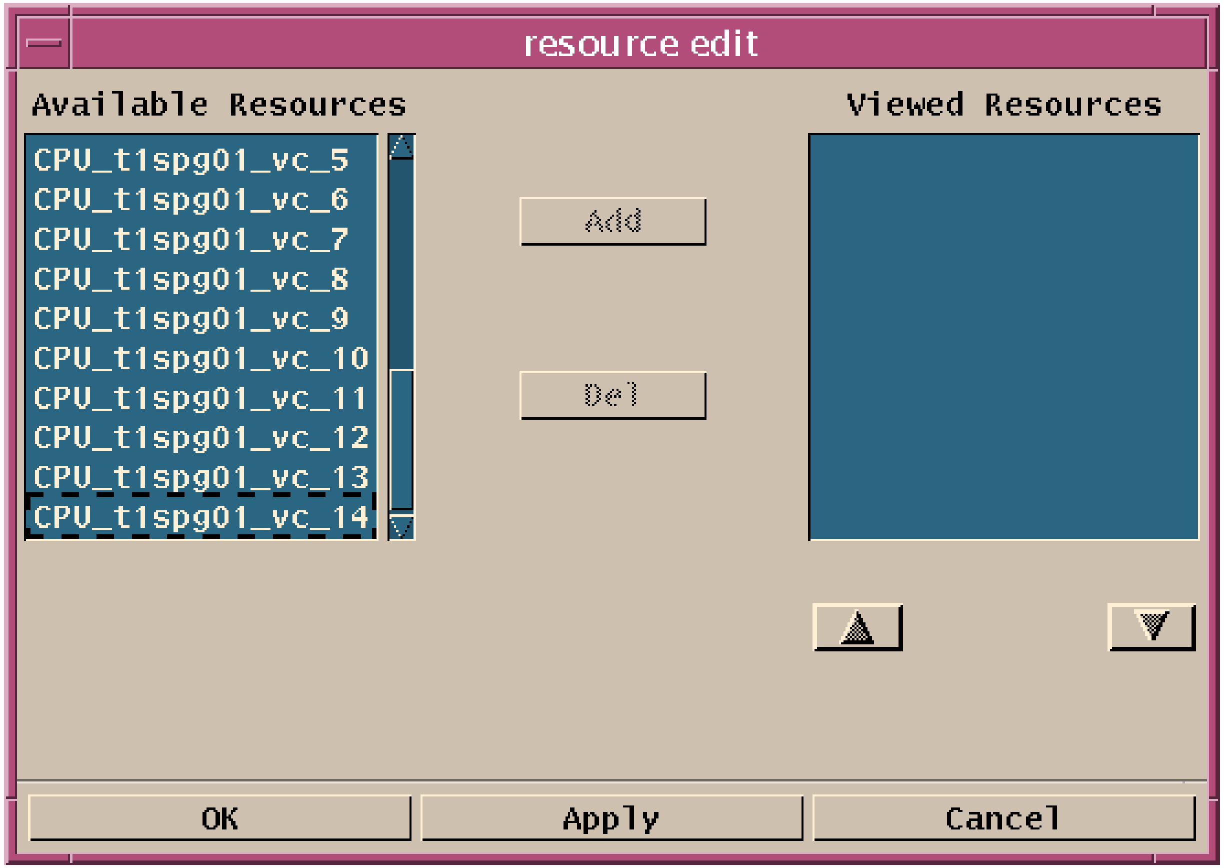
Planning Timeline
Open Plan Window



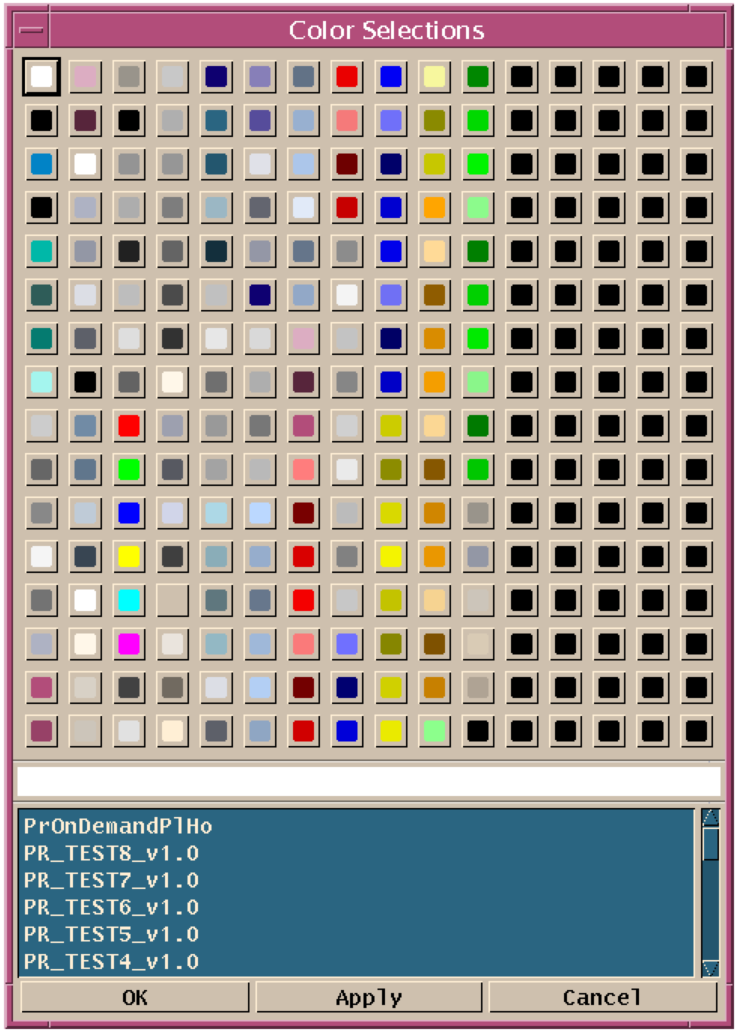
Planning Timeline
Plan Window Edit Window



Planning Timeline: Resource Edit Window



Planning Timeline Color Selections Window



Cleaning the PDPS Database and Science Processing Disks

- **At DAACs with a heavy data processing load it is essential to perform PDPS database cleanup and virtual-computer "garbage collection" on a regular basis**
  - **Frequency of cleanup is determined by each DAAC's needs**
  - **Failure to perform cleanup is likely to lead to performance problems with the Planning and Data Processing Subsystems**

Cleaning the PDPS Database and Science Processing Disks

- Recommendation is that the database cleanup and “garbage collection” processes be run in the following order:
  - Database cleanup
  - “Garbage collection” on the virtual computer(s)
  - Database cleanup (again)
- If not already done, it is possible to create a script to run the preceding processes
  - If necessary, processes can be run individually
- To the extent possible cleanup scripts should be run when the system is relatively quiet:
  - No PRs/DPRs being created
  - No jobs running in AutoSys

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- **Cleaning the PDPS Database (EcPlDbClean script)**
  - **In the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Planning/Management Workstation**
  - **Cleans up some tables in a PDPS database**
  - **Tries to delete applicable records in the following order:**
    - **Data Processing Requests based on timeStamp and completionState(SUCC_DEL)**
    - **Production Requests that have no associated DPRs**
    - **Dynamic data granules that are not used by any DPR or by the Data Processing Subsystem**
    - **PGEs that are marked with a deleteFlag**
    - **Science Software that has no associated PGE**
  - **Compiles a list of data granules that are not deleted because the Data Processing Subsystem needs to use them**

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- **Cleaning the PDPS Database: Procedure**
  - **Access a terminal window logged in to the Planning/Management Workstation host**
  - **Change directory to the subdirectory (e.g., utilities) containing the production planning startup scripts**
  - **Start the EcPlDbClean script using the following arguments:**
    - ***MODE***
    - ***dbuser* (user name for logging in to interactive structured query language (isql))**
    - ***dbpassword* (password for isql login)**
    - ***dbserver* (name of the PDPS database server)**
    - ***months* (number specifying the removal of records that are older than that number of months)**
    - ***days* (optional) (number that specifies the removal of records that are older than that number of days)**

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- Performing Garbage Collection
  - EcDpPrGarbageCollectorStart on the Queuing Server
    - Run to delete unneeded files from the science processing disks and update the PDPS database accordingly
    - Actual executable invoked by the script is EcDpPrDeletionClient
  - EcDpPrGarbageCollectorStart differs from EcDpPrDeletionClientStart in the following ways:
    - Does not open a separate xterm window
    - Requires specification of a MACHINE_TO_COLLECT variable
    - Includes retry logic in case of database deadlock

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- **Performing Garbage Collection (Cont.): Procedure**
  - **Access a terminal window logged in to the Queuing Server host**
  - **Change directory to the subdirectory (e.g., utilities) containing the production processing startup scripts**
  - **Start the EcDpPrGarbageCollectorStart script using the following arguments:**
    - ***MODE***
    - ***machine* - virtual computer (e.g., x0spg01_vc) for which garbage collection is being requested**
    - ***retries* - number of retries in case of database deadlock**
    - ***interval* - amount of time (in minutes) between retries**
  - **Observe the results as the script runs**

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- **Running the Deletion Server Client**
  - **EcDpPrDeletionClientStart script on the Queuing Server**
    - **Run to delete unneeded files from the science processing disks and update the PDPS database accordingly**
    - **Executable invoked by the script is EcDpPrDeletionClient**

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- **Running the Deletion Server Client (Cont.): Procedure**
  – **Access a terminal window logged in to the Queuing Server host**
  – **Change directory to the subdirectory (e.g., utilities) containing the production processing startup/utility scripts**
  – **Start the EcDpPrDeletionClientStart script using the following argument:**
    - ***MODE***
  – **Observe the results as the script runs**

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- **Resolving PDPS Database and Science Processing Disk Content Discrepancies**
  - **EcDpPrRmFilesWOGranules.pl script on the Queuing Server**
    - **Is run to ensure consistency between the file references in the PDPS database and the files actually staged on the science processing disks**
    - **Generates a list of files with consistent references among tables in the PDPS database**
    - **Checks for files on the disk that are not included in the list of files referenced in the PDPS database and either lists the inconsistent files or generates a script to delete them (as requested)**
    - **Checks to determine whether the disk partitions referenced in the PDPS database actually exist on the disk(s)**
    - **Removes all file references in the PDPS database that are not included in the list of files**

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- **Resolving PDPS Database and Science Processing Disk Content Discrepancies (Cont.)**
  - **The EcDpPrRmFilesWOGranules.pl script should be run when the system is relatively quiet**
    - **When no jobs are running in AutoSys**

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- **Resolving PDPS Database and Science Processing Disk Content Discrepancies: Procedure**
  - **Access a terminal window logged in to the Queuing Server host**
  - **Change directory to the subdirectory (e.g., utilities) containing the production processing startup/utility scripts**
  - **Start the EcDpPrRmFilesWOGranules.pl script using the following arguments:**
    - ***dbuser***
    - ***dbpassword***
    - ***MODE***
    - ***dbserver***
    - **fix [or nofix]**
  - **Observe the results as the script runs**

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- **Saving and/or Resetting the PDPS Database**
  - **Scripts used in saving and resetting the database**
    - **EcPlDbReset**
    - **EcPlDbList**
    - **EcPlDbSave**
  - **Saving the database (using EcPlDbSave) produces one ASCII file (with a ".dat" extension) for each database table**
  - **Resetting the database involves clearing ("wiping out") the data in the database tables and loading values from a specified "saved database" file**

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- **Saving and/or Resetting the PDPS Database (Cont.)**
  - **It is important to take into consideration the consequences of resetting the database before performing the procedure**
    - **Removes and replaces all Resource Definitions, Resource Reservations (Ground Events), Production Requests, Data Processing Requests, and Production Plans**
    - **Coordinate with all affected parties, including the Resource Planner, Production Planner, and Production Monitors concerning the effects of resetting the database as well as its after-effects (e.g., recreating resource definitions, resource reservations, and production requests)**

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- Saving and/or Resetting the PDPS Database (Cont.)
  - Whenever the PDPS database is reset (not including database cleanup using the EcPlDbClean script) it is also necessary to remove all PLS subscriptions in the Communications Subsystem (CSS) Subscription Server database
    - Production personnel can remove the subscriptions using the Subscription Server GUI (EcSbSubServerGUI) if they have access to the GUI
    - Otherwise, they can request User Services personnel to remove the subscriptions
    - As a result of removing the subscriptions, no subscription notification will come through for existing jobs in the newly loaded database
    - Only new jobs generated using the Production Request Editor will work normally with regard to subscriptions

Cleaning the PDPS Database and Science Processing Disks (Cont.)

- **Saving and/or Resetting the PDPS Database: Procedure**
  - NOTE: It is important to log in as a user who has “write” permission in the saved_dumps directory; otherwise it will not be possible to save database contents
  - Access a terminal window logged in to the Planning/Management Workstation host
  - Set the ECS_HOME environmental variable if necessary
  - Change directory to the subdirectory (e.g., utilities) containing the production planning startup scripts
  - If desired, save the database
  - If desired, obtain a listing of saved databases
  - If desired, reset the database

Troubleshooting Production Planning Problems

- **Troubleshooting:**

  process of identifying the source of problems on the basis of observed trouble symptoms

Troubleshooting Production Planning Problems (Cont.)

- **Problems with production planning can usually be traced to…**
  - **some part of the Planning Subsystem**
  - **problems in other subsystems, including (but not necessarily limited to):**
    - **Data Processing Subsystem (DPS)**
    - **Data Server Subsystem (DSS)**
    - **Communications Subsystem (CSS)**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery**
  - Each request that crosses a client/server boundary is assigned a system-unique identifier referred to as an RPC ID
  - The RPC ID facilitates the automatic fault recovery events that occur whenever there is a client or server failure
  - As a request propagates through the system, each associated client/server exchange is assigned a unique RPC ID
    - The RPC ID for each interaction is derived from the previous RPC ID received by the client for the request; consequently, all RPC IDs associated with a given request have a common portion that relates the various client/server calls to one another
    - Given the previous RPC ID, clients consistently reproduce the same RPC ID that was submitted to the server on the subsequent event

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery (Cont.)**
  - The concept of reproducible RPC IDs is central to the system fault recovery capability
    - When requests are retried from client to server, they are always submitted with the same RPC ID that was used in the original submission of the request, even if either client or server has crashed between retries
  - The RPC ID is also central to the check-pointing aspect of fault recovery
    - As requests arrive at fault recovery-enabled servers, they are recorded in a persistent store (typically a database), tagged with the RPC ID
    - As the request is serviced, check-pointing state information may be updated in the persistent store, up to and including the request's completion status
    - This allows the servers to resume servicing from the last check-pointed state, particularly upon resubmission from a client

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery (Cont.)**
  - PLANG and PRONG components check-point the following types of information:
    - EcPlSubMgr - Unprocessed subscription notifications
    - EcDpPrDeletion - Interim Delete Requests
    - EcDpPrEM - Queued and activated jobs

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Fault Handling**
  - Failure events are classified according to the following three severity levels:
    - Fatal error
    - Retry error
    - Warning

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Fault Handling (Cont.)**
  – Fatal error is returned when a request cannot be serviced, even with operator intervention
    - For example, if a request is made to distribute data via ftp to a non-existent host, the request is failed
  – Retry error is a potentially recoverable error
    - Normally, a retry error would be returned to the client only when the server cannot recover from the error automatically
    - A retry error may require operator assistance
    - For example, entering a new name for a PR after being notified that a previously entered name contained too many characters
  – Warning is provided when operations can proceed but an unexpected circumstance was detected
    - For example, if a client requests removal of a file but the file does not exist

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Fault Handling (Cont.)**
  - **Transient errors (such as network errors) are always retry errors**
    - **In general, clients and servers that experience transient retry errors first attempt to recover by retrying the operation automatically**
    - **One special case of this is “rebinding,” which refers to the process by which a client automatically attempts to re-establish communication with a server in the event communication is disrupted**
    - **The disruption may be caused by transient network failure, or by the server crashing or being brought down**
    - **In any case, the client automatically attempts to reconnect to the server for a configurable period of time on a client-by-client basis**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Fault Handling (Cont.)**
  - System processes encountering an error or receiving an error from a server request can either pass the error back to a higher-level client or present it to the operator for operator intervention
  - The specific fault handling policies for PLANG and PRONG client processes are shown in the table that follows

Troubleshooting Production Planning Problems (Cont.)

PLANG and PRONG Fault Handling Policies

Client Process	Fault Handling Policy
EcPlSubMgr	**Retry errors:** All Subscription processing errors are retried a configurable number of times and for a configurable time period. After the configurable number of times (or time period) the subscription is lost. **Fatal errors:** N/A.
EcPlPREditor_IF **EcPlWb**	**Retry errors:** Since these are GUI applications, errors are reported to the operator and it is the operator’s responsibility to retry the request. **Fatal errors:** Errors are reported to the operator.
EcPlOdMgr	**Retry errors:** Retries errors from the Science Data Server and the Subscription Server. **Fatal errors:** Logs errors and stops current on demand requests.
EcDpPrEM	**Retry errors:** Errors are retried a configurable number of times, then the job is failed and it is up to the Production Monitor to restart the job through AutoSys. **Fatal errors:** A fatal error message is logged.
EcDpPrJobMgmt	**Retry errors:** If a DPR cannot be assigned to a machine or created in AutoSys, it is left in a PENDING state and the assignment is retried after DpPrPendingThreadWaitInterval seconds. **Fatal errors:** N/A.

Troubleshooting Production Planning Problems (Cont.)

PLANG and PRONG Fault Handling Policies (Cont.)

Client Process	Fault Handling Policy
EcDpPrDeletion	**Retry errors:** No retries are implemented. Status from DSS is not checked. **Fatal errors:** N/A.

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Client Crash and Restart**
  - When a client of a PLANG or PRONG server crashes, the server (i.e., EcPlSubMgr, EcDpPrJobMgmt, or EcDpPrDeletion) continues to service the requests that were in process at the time of the client's crash
  - When a client restarts in the system, it sends a restart notification to each server with which it interacts
    - Clients notify servers that they have come up either "cold" or "warm"
    - Generally, the notification temperature sent to the server matches the temperature at which the client process is restarted

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Client Crash and Restart (Cont.)**
  - **Default server behavior in response to "warm" startup notification from a client:**
    - **Outstanding requests for the restarted clients remain available in the persistent store**
    - **The outstanding requests may be resubmitted by the client, and are serviced to completion upon resubmission**
    - **Associated resources are left allocated until the requests are completed**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Client Crash and Restart (Cont.)**
  - Default server behavior in response to “cold” startup notification from a client:
    - All outstanding requests for the restarted client are cancelled
    - If the client resubmits any cancelled request using the same RPC ID (e.g., by pressing the Retry button from an operator GUI), it is failed with a fatal error due to the client cold startup notification
    - Any resources associated with the cancelled requests are released and reclaimed by the system

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Server Crash and Restart**
  - When a server crashes, clients cannot continue to submit requests for processing
  - Synchronous requests in progress result in a Distributed Computing Environment (DCE) exception being thrown back to the client process, which enters a rebinding failure recovery mode (as previously mentioned)
  - Attempts to submit requests while the server is down result in the client blocking until a communication timeout has been reached
  - Although DCE has been replaced by socket-based library calls (i.e., CCS Middleware), the DCE exception code is handled by the CCS Middleware

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Server Crash and Restart (Cont.)**
  - When a server restarts, it may perform various resynchronization activities in order to recover from an unexpected termination
  - In the event of a server cold start or cold restart, the server typically cancels all outstanding requests and reclaims all associated resources
  - In general, existing request queues are retained for warm restarts and cleared for cold starts or cold restarts

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Server Crash and Restart (Cont.)**
  - **EcPlSubMgr-specific activities upon start/restart:**
    - **Warm Restart: Any subscriptions that have not been processed are read from checkpoint file and processed**
    - **Cold Start or Cold Restart: N/A**
  - **EcDpPrJobMgmt-specific activities upon start/restart:**
    - **Warm Restart: Jobs in AutoSys and jobs waiting in the queue are read from the database; any jobs that are ready are placed into AutoSys from the queue (if there are processing slots available)**
    - **Cold Start or Cold Restart: N/A**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Server Crash and Restart (Cont.)**
  - **EcDpPrDeletion-specific activities upon start/restart:**
    - **Warm Restart: Interim granules marked for deletion are read from the database and are deleted when time-out occurs**
    - **Cold Start or Cold Restart: N/A**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Request Resubmission**
  - Upon restarting a crashed client or server, requests are typically resubmitted
  - If the restarted process was started warm, the fault-recovery capabilities permit the server to resume processing of the request from its last check-pointed state
    - This prevents needless repetition of potentially time-consuming activities
  - EcDpPrJobMgmt- and EcDpPrDeletion-specific activities upon resubmission of a request:
    - Requests are submitted synchronously
    - If the entire request is resubmitted by a client, then only that part of the resubmitted request that has not been completed is reprocessed

Troubleshooting Production Planning Problems (Cont.)

- **Troubleshooting table**
  - describes actions to be taken in response to some common Production Planning problems
  - if the problem cannot be identified and fixed without help within a reasonable period of time, call the help desk and submit a trouble ticket in accordance with site Problem Management policy

Troubleshooting Production Planning Problems (Cont.)

Symptom	Response
Unable to log in to the Planning Subsystem host (e.g., e0pls03).	Check with the Operations Controller/System Administrator to ensure that the host is “up."
GUI not displayed when the start-up script has been properly invoked.	Ensure that the DISPLAY variable was set properly. [For detailed instructions refer to the applicable procedure, either **Launching the Production Request Editor** or **Launching Planning Workbench-Related GUIs** (previous sections of this lesson).]
Error message indicating that SNS (System Name Server) and/or Resource Model is/are in use using the selected Application ID.	1. Use another Application ID if working in a different mode from the person using the selected Application ID. 2. If working in the same mode as the other user, coordinate use of Planning applications with the other user and/or the System Administrator. [For detailed instructions refer to the procedure for **Launching Planning Workbench-Related GUIs** (previous section of this lesson).]
Error message associated with the Production Request Editor.	Refer to Table 3, Production Request Editor User Messages (adapted from the corresponding table in 609-EMD-001, *Release 7.10 Operations Tools Manual for the EMD Project*).
Error message associated with the Production Strategies GUI.	Refer to Table 4, Production Strategy User Messages (adapted from the corresponding table in 609-EMD-001, *Release 7.10 Operations Tools Manual for the EMD Project*).

Troubleshooting Production Planning Problems (Cont.)

Symptom	Response
Error message associated with the Planning Workbench.	Refer to Table 5, Planning Workbench User Messages (adapted from the corresponding table in 609-EMD-001, *Release 7.10 Operations Tools Manual for the EMD Project*).
Production Request fails (DPR generation fails).	1. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 2. If hosts/servers are all "up," perform the procedure for **Handling a Failure to Generate DPRs** (subsequent section of this lesson). 3. Retry generating DPRs by resaving the Production Request. [For detailed instructions refer to the section on **Editing/Modifying a Production Request** (previous section of this lesson).]

Troubleshooting Production Planning Problems (Cont.)

Symptom	Response
PR or DPR deletion hangs.	1. Ensure that enough time has passed to allow DPR deletion (deleting a DPR can require as much time as creating a DPR). 2. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). (Both the Job Management Server and Deletion Server are called to clean up all PDPS database tables associated with the DPR or PR.) [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 3. If hosts/servers are all "up," check for a database lock or resource lock in the PDPS database. [For detailed instructions refer to the section on **Responding to PR or DPR Deletion that Hangs** (subsequent section of this lesson).]

Troubleshooting Production Planning Problems (Cont.)

Symptom	Response
DPR deletion fails.	1. Ensure that enough time has passed to allow DPR deletion (deleting a DPR can require as much time as creating a DPR). 2. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). (Both the Job Management Server and Deletion Server are called to clean up all PDPS database tables associated with the DPR or PR.) [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 3. If hosts/servers are all "up," check the Deletion Server Debug log (EcDpPrDeletionDebug.log). [For detailed instructions refer to the section on **Responding to DPR Deletion that Fails** (subsequent section of this lesson).]

Troubleshooting Production Planning Problems (Cont.)

Symptom	Response
DPR scheduling fails (DPR is not passed to Data Processing).	1. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 2. If hosts/servers are all "up," perform the procedure for **Handling a DPR Scheduling Failure** (subsequent section of this lesson). 3. If necessary, delete the affected DPRs. [For detailed instructions refer to the section on **Creating a New Production Plan** (previous section of this lesson).] 4. If affected DPRs were deleted, recreate the DPRs. [For detailed instructions refer to the section on **Editing/Modifying a Production Request** (previous section of this lesson).] 5. If affected DPRs were recreated, create a new production plan. [For detailed instructions refer to the section on **Creating a New Production Plan** (previous section of this lesson).]

Troubleshooting Production Planning Problems (Cont.)

Symptom	Response
Other problems.	1. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 2. If hosts/servers are all “up,” check the log files (e.g., EcPlPREditor.ALOG, EcPlPREditorDebug.log, EcPlWb.ALOG, EcPlWbDebug.log, and EcPlTl.ALOG) in the /usr/ecs/*MODE*/CUSTOM/logs directory for error messages. [For detailed instructions refer to the procedure for **Checking Log Files** (subsequent section of this lesson).]

Hosts, Servers, etc. Relevant to Planning & Processing

HOST	SERVER/CLIENT/OTHER SOFTWARE
Planning/Management Workstation (e.g., x0pls01)	Production Request Editor (EcPlPREditor_IF) Planning Workbench GUI (EcPlWb) Production Strategies GUI (EcPlProdStrat) Production Planning Master Timeline (EcPlTl) Message Handler (EcPlMsh) System Name Server (EcPlSns) Resource Model (EcPlRm) Production Request (PR) Generator (EcPlPRGenerator)
Queuing Server (e.g., x0sps04)	Job Management Server (EcDpPrJobMgmt) Deletion Server (EcDpPrDeletion) Execution Management (EcDpPrEM) AutoSys Event Processor (event_demon) AutoSys Event Server (Sybase server) (e.g., x0sps02_srvr) On-Demand Manager (EcPlOdMgr) Subscription Manager (EcPlSubMgr) PDPS database Sybase server (e.g., x0sps02_srvr)
Science Processor (e.g., x0spg11)	PGE Management (EcDpPrRunPGE) Resource Usage (EcDpPrRusage) PGEs

Hosts, Servers, etc. Relevant to Planning & Processing (Cont.)

HOST	SERVER/CLIENT/OTHER SOFTWARE
Access/Process Coordinators (APC) Server (e.g., x0acg01)	Archive Server (EcDsStArchiveServer) FTP Server (EcDsStFtpServer) Cache Manager Server (EcDsStCacheManagerServer) Staging Disk Server (EcDsStStagingDiskServer) Pull Monitor Server (EcDsStPullMonitorServer)
Ingest Server (e.g., x0icg01)	Name Server (EcCsIdNameServer) Registry Server (EcCsRegistry)
Sun Consolidation External Server (e.g., x0ins01)	Data Dictionary (EcDmDictServer)
Sun Consolidation Internal Server (e.g., x0acs11)	Science Data Server (EcDsScienceDataServer) Subscription Server (EcSbSubServer)

Troubleshooting Production Planning Problems (Cont.)

- **Production Planning Troubleshooting Procedures**
  - Checking Connections to Hosts/Servers
  - Checking the Production Request Editor ALOG File
  - Using ISQL to Check Database Tables
  - Checking the PDPS Database for Causes of Failure to Generate DPRs
  - Checking the Production Request Editor Debug File for Evidence of Metadata Queries
  - Checking for Database Deadlocks
  - Checking for Resource Locks in the PDPS Database

Troubleshooting Production Planning Problems (Cont.)

- Production Planning Troubleshooting Procedures (Cont.)
  - Responding to DPR Deletion that Fails
  - Responding to a "DPR Validation Failed" Error
  - Responding to an "information (INFO) Production Request {Production Request Id} has unschedulable DPR {DPR Id}" Error
  - Checking Log Files
  - Checking Database Connections

Troubleshooting Production Planning Problems (Cont.)

- **Procedure (Checking Connections to Hosts/Servers):**
  - **Access a terminal window logged in to the Planning/Management Workstation host**
  - **Change directory to the utilities subdirectory (/usr/ecs/*MODE*/CUSTOM/utilities)**
  - **At the command line prompt enter EcCsIdPingServers *MODE***
  - **Observe the results displayed on the screen to determine whether connections can be made with the necessary hosts and servers**
  - **Ping the servers again (EcCsIdPingServers *MODE*)**
  - **If it is not possible to connect to any needed host(s)/server(s), notify the Operations Controller/System Administrator to check the hosts/servers and bring them back up if necessary**

Troubleshooting Production Planning Problems (Cont.)

- Handling a Failure to Generate DPRs
  - Checking the Production Request Editor ALOG File
  - Using ISQL to check database tables
  - Checking the PDPS database for causes of failure to generate DPRs
  - Determining whether DPR explosion fails because Production Request Editor does not query DSS for data
  - Checking the Production Request Editor debug file for evidence of metadata queries
- Responding to PR or DPR Deletion that Hangs
  - Checking for database deadlocks
  - Checking for resource locks in the PDPS database

Troubleshooting Production Planning Problems (Cont.)

- Responding to DPR Deletion that Fails
- Handling a DPR Scheduling failure
  - Responding to a "DPR Validation Failed" error
  - Responding to an "information (INFO) Production Request {Production Request Id} has unschedulable DPR {DPR Id}" error
- Checking Log Files
- Checking Database Connections

Production Processing

- DPR Chaining
  - The Job Management Server in the DPS schedules chains of DPRs through the use of stored procedures in the PDPS database
  - In addition, when managing chains of DPRs, Job Management uses data from the following two tables in the PDPS database:
    - DpPrPgeLimits
    - DpPrClassSchedulingLimits

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - DpPrPgeLimits imposes restrictions on the number of chain-head DPRs of a particular PGE that can run simultaneously on the same virtual machine
    - A database record defines each PGE-virtual computer combination that will be run
  - DpPrPgeLimits table has the following columns:
    - pgeId - PGE ID
    - computerName – virtual computer (vc) name
    - maxConcurrent - defines the maximum number of chain heads with a particular PGE ID that can run at the same time on a specific vc
    - numConcurrent - shows how many chain heads with a particular PGE ID are running at the same time on a specific vc
    - numScheduled - shows how many chain heads with a particular PGE ID are currently scheduled on a vc

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - If there were no record in the DpPrPgeLimits table for a particular PGE-computer combination that was scheduled, there would be no limits on the DPRs for that PGE
    - PGE would be able to run and there would be no limits placed on how many DPRs for that PGE could run on the same virtual machine
    - DPRs would run on the machine specified by the Production Planner in the Production Request (if the Production Planner designated a machine when creating the Production Request)
    - If no machine was specified in the Production Request, the machine would be determined from the PlResourceRequirement table (data entered during PGE registration)

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - DpPrClassSchedulingLimits limits the classes of DPRs that can run at any point in time
    - Classes correspond to the types of processing
  - DpPrClassSchedulingLimits table has three records, one for each type of processing
  - Each record has the following fields:
    - dprClass - assigned value identifies the type of processing (0 = Routine Processing, 1 = On-Demand Processing, 2 = Reprocessing)
    - maxDprs - maximum number of jobs of the type (in dprClass) that are allowed to run on the system
    - minDprs - currently not used
    - currentDprs - number of jobs of the type (in dprClass) that are currently running

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - If the DpPrClassSchedulingLimits table has no record for a particular type of processing, DPRs of that type are not allowed into AutoSys
  - Values for the maxDprs and minDprs columns in the DpPrClassSchedulingLimits table are loaded at Job Management Server startup using data from two configuration parameters:
    - DpPrMaxConcurrentDPRs - maximum allowed jobs
    - DpPrMinConcurrentDPRs - minimum allowed jobs
  - Each parameter has three integer values; the first for routine processing; the second for on-demand processing; and the third for reprocessing jobs

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - Example: Configuration Registry may have the following entries:
    - DpPrMaxConcurrentDPRs = 100 60 40
    - DpPrMinConcurrentDPRs = 0 0 0
  - In this case the maximum allowed jobs is 100 for routine processing, 60 for on-demand processing, and 40 for reprocessing
  - Minimum allowed jobs is 0 for each type of processing

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - Total number of completed jobs allowed in AutoSys is defined by a configuration parameter:
    - DpPrAutoSysMaxDPRs
  - Within the restrictions of the DpPrClassSchedulingLimits and DpPrPgeLimits database tables classes of DPR chains are scheduled by DPS with chains of DPRs (identified by their chain heads) being scheduled on the same machine whenever possible
  - When scheduling a chain-head DPR, attempts are made to schedule it on the machine that has the highest number of accepted inputs if accepted inputs are found
  - The chain-head DPR's children should be scheduled on the same machine

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - Load balancing is done among virtual computers on the same string when selecting a computer for a DPR chain
  - If a DPR does not have its PGE identified in the DpPrPgeLimits table, that does not prevent it from being scheduled on a particular machine if most of the DPR's inputs are staged on that machine
  - If there is more than one chain head for a set of DPRs, the DPRs are combined into a single chain
  - For example:
    - DPR#1 and DPR#2 are both parents of DPR#3
    - Marking DPRs#1 and #2 as chain heads results in a single chain consisting of all three DPRs

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - Another example (MODIS processing:
    - If the Production Planner defines both the MODPGE01 PR and the MODPGE02 PR as chain heads, they are combined in one chain with the MODPGE01 DPR ID as the chainId
    - MODPGE01 is designated the chain head because its outputs are used as inputs to MODPGE02 and MODPGE08
    - MODPGE02 outputs are used as input to MODPGE08
  - The PGEs that will be used as chain heads should be identified before installation
  - In selecting potential chain heads, it is recommended that PGEs which create substantial (i.e., large or many) inputs for other PGEs be so designated

Production Processing (Cont.)

- Copy on Demand Feature
  - A feature whereby the DPS code stages granules locally; i.e., granule files are staged to the processor on which the PGE is running
  - The copy on demand feature is specified in the following locations:
    - onDemandCopy column in the PlDataTypeMaster database table (PDPS database) (the value of the onDemandCopy flag is used by the system to determine whether input files need to be locally staged)
    - ON_DEMAND_COPY flag in the ESDT ODL files after DATA_TYPE and DATA_TYPE_VERSION; for example:

      ON_DEMAND_COPY = “Y”

Production Processing (Cont.)

- **DPR Output Files Immediately Available as Input**
  - **Output files are immediately available as input when generated as output of a DPR**
  - **There is no waiting for Subscription Server notification before the output of a parent DPR is used as input to a child DPR**

Production Processing (Cont.)

- **AutoSys**
  - **production scheduling tool**
  - **supports the operational activities surrounding production processing in the PDPS**
  - **assists with the following activities (among others):**
    - **job monitoring**
    - **job scheduling**
    - **fault notification**
    - **job restart**
    - **determining the effects of failure of a DPR**
    - **determining the cause and actions to be taken due to the failure of a DPR**

Production Processing (Cont.)

- **AutoSys (Cont.)**
  - **displays DPRs as job boxes**
  - **recognizes the following categories of jobs:**
    - **box jobs**
    - **command jobs**
    - **file-watcher jobs**

Production Processing (Cont.)

- Box job
  - – collection of other jobs
  - – provides an organizational structure for a group of jobs that should be run within the same time period
    - - performs no processing action
- Command job
  - – "command" can be a shell script, the name of an executable program, a file transfer, or any other command that causes execution of a UNIX command on client machine

Production Processing (Cont.)

- File-watcher job
  - functions similarly to a command job
  - monitors the creation and size of a particular operating system file
  - allows AutoSys to know the status of external files that are needed in the processing of command jobs or box jobs

Production Processing (Cont.)

- AutoSys Job Starting Parameters
  - Date and time scheduling parameters are met
  - Starting Conditions specified in the job definition evaluate to “true”
  - For jobs in a box, the box must be in the RUNNING state
  - The current status of the job is not ON_HOLD or ON_ICE
- AutoSys finds all jobs that may be affected by any change in the truth of the starting parameters and determines whether or not to start the jobs

Production Processing (Cont.)

- **AutoSys Jobs**
  - Each DPR generated by the Planning Subsystem defines a box job for AutoSys
  - Every DPR/box job is composed of three command jobs that run in the following order:
    - Preprocessing (EcDpPrEM)
    - Execution (EcDpPrRunPGE)
    - Postprocessing (EcDpPrEM)
  - The number of command jobs in an AutoSys Job box was reduced from seven to three in order to reduce overhead
    - Reduces the load on the AutoSys Event Processor by about one half
    - Has no effect on the DPS Queuing Server CPU loading

Production Processing (Cont.)

- AutoSys Jobs
  - To create the new Preprocessing job the following old command jobs were combined:
    - Allocation
    - Staging
    - Pre-processing
  - To create the new Postprocessing job the following old command jobs were combined:
    - Post-processing
    - Destaging
    - Deallocation
  - The value in the database column dprState correlates to the seven old job states

Production Processing (Cont.)

- **Job Names**
  - **Indicate the PGE and the mode in which the DPR was generated and the stage of processing**
  - **Example:**
    - **AM1Eph#2.012302200TS2**
    - **AM1Eph#2.012302200TS2R**
    - **AM1Eph#2.012302200TS2E**
    - **AM1Eph#2.012302200TS2P**

Production Processing (Cont.)

- **Job Names (Cont.)**
  - The first job name in the list is a DPR/box job-level name
    - First few characters identify the PGE (i.e., AM1Eph - Step 1 DPREP)
    - Last three characters of the DPR/box job-level name (i.e., TS2) indicate the mode in which the DPR was generated
    - Last four characters of the remaining (command) job names in the list indicate the mode (i.e., TS2) and the stage of processing
    - Job name ending in “R” is the preprocessing job; job name ending in “E” is the execution job; job name ending in “P” is the postprocessing job

Production Processing (Cont.)

- **Preprocessing depends on the box job having started**
- **Execution depends on successful completion of the command job that precedes it (preprocessing)**
- **Postprocessing depends on Execution having completed (not necessarily "successful," just done)**
  - **If execution failed, postprocessing handles failed PGE processing**

Production Processing (Cont.)

- DPR (box job) may be dependent on the successful completion of some other DPR
  – usually involving a need for the output of another DPR as input
- Effects of DPR dependencies
  – dependent DPRs are "held" by AutoSys until their data availability subscriptions are fulfilled
  – subscription manager software (in the PLS) informs the DPS to release the AutoSys jobs after all data subscriptions for a given DPR are fulfilled
  – DPS (as monitored by the AutoSys Job Scheduling engine) runs the PGEs and associated jobs as the resources required for the tasks become available
  – procedure continues until all DPRs scheduled for the day have completed

Production Processing (Cont.)

- Each mode has multiple Job Management queues
  - One queue for each type of processing
  - There is a maximum number of jobs allowed in AutoSys for each type of processing
  - For example, in each mode a DAAC would have queues for the following types of processing:
    - Routine Processing
    - On-Demand Processing
    - Reprocessing

Production Processing (Cont.)

- **DAAC Production Monitor uses AutoSys/Job Management Web Interface when monitoring and controlling job processing, including the following functions:**
  - **Deleting/suspending/resuming jobs as required**
  - **Monitoring and providing processing status upon request**

Launching the AutoSys GUI Control Panel

- **Production Processing Applications**
  - **Subscription Manager**
  - **Job Management**
  - **Execution Management**
  - **PGE Management**
  - **Deletion Server**
  - **AutoSys**
    - **Event Processor**
    - **Event Server**
    - **AutoSys GUIs**
    - **Job Management Web Interface**

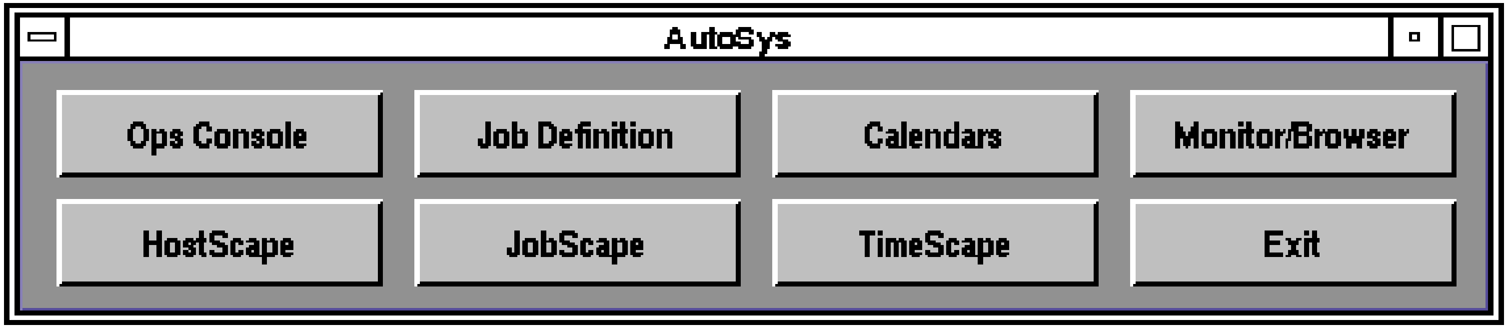
Launching the AutoSys GUI Control Panel

- **Production Processing Applications (Cont.)**
  - **QA Monitor**
  - **Sybase ASE Server**

Launching the AutoSys GUI Control Panel (Cont.)

- **Procedure**
  - Access a terminal window logged in to the Queuing Server host
  - Set the ECS_HOME environmental variable if necessary
  - Change directory to the subdirectory (e.g., autouser) containing the set-up files
  - Source the appropriate file
  - Change directory to the subdirectory (e.g., utilities) containing the AutoSys start script
  - Start AutoSys in the appropriate mode

AutoSys GUI Control Panel



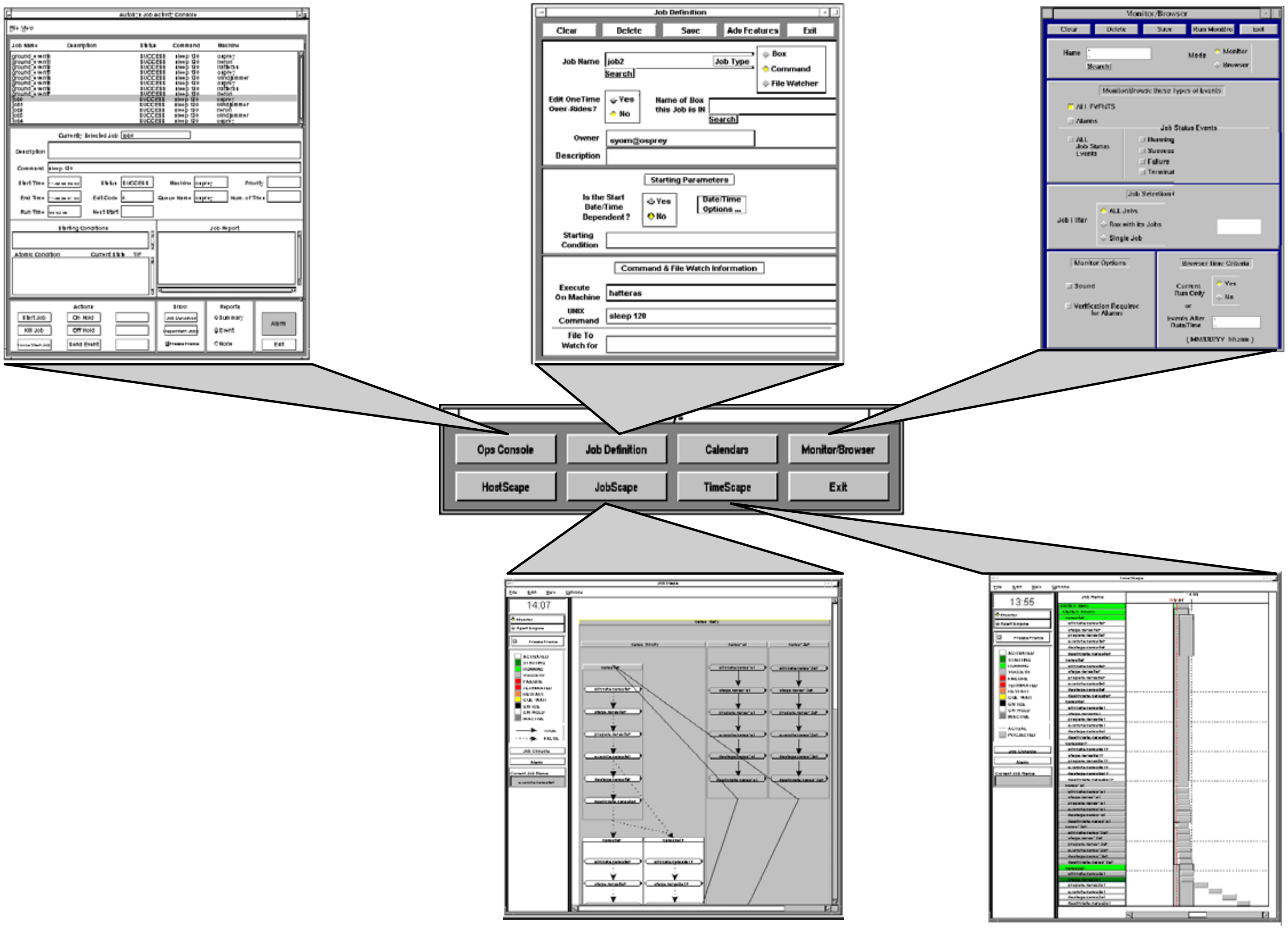
Configuring AutoSys Screens/Displays

- **Configuring AutoSys Runtime Options**
  - **Refresh Interval**
    - **determines how often the View Region will be updated**
  - **Inches/Hr**
    - **indicates how much information is displayed on the screen**

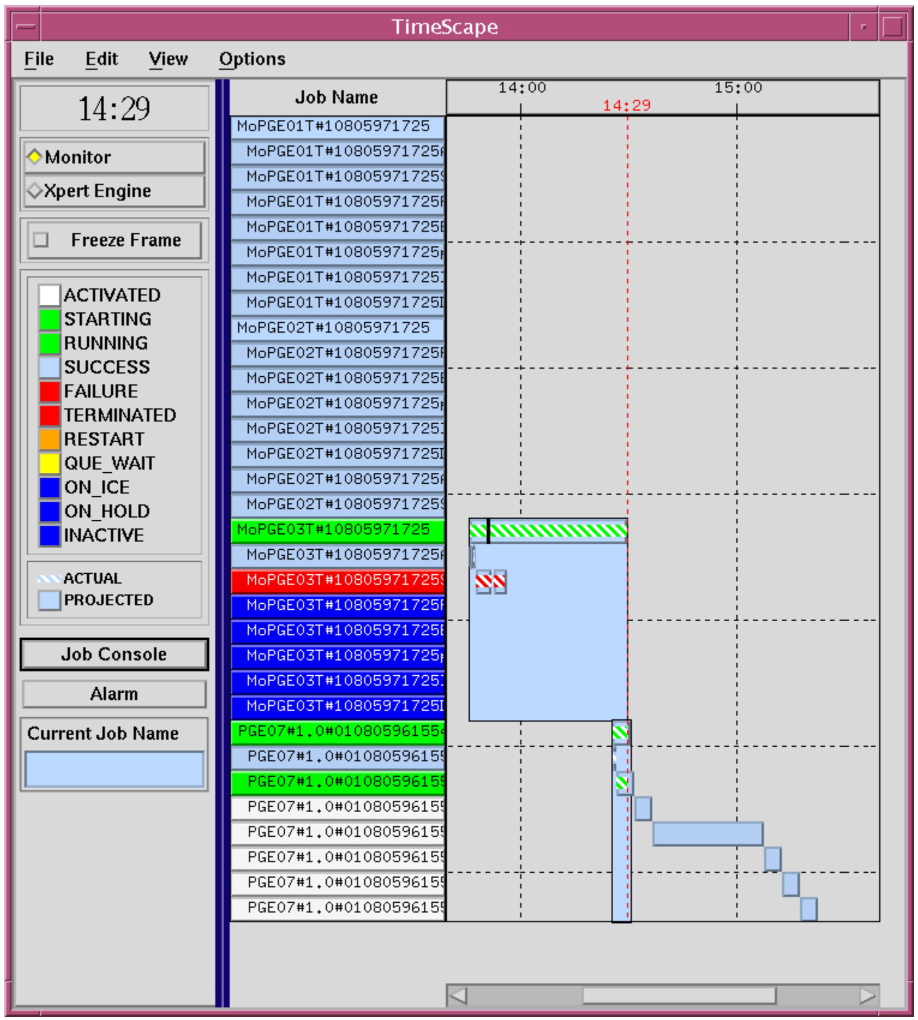
Configuring AutoSys (Cont.): Configuring Runtime Options

- **Procedure**
  - Click on either the TimeScape or JobScape button to display the corresponding interface
  - Select Options → Edit Runtime Options from the pull-down menu to display the Runtime Options dialog box
  - Enter new values for the runtime options as necessary
    - refresh interval
    - number of inches/hour
  - Apply the modifications

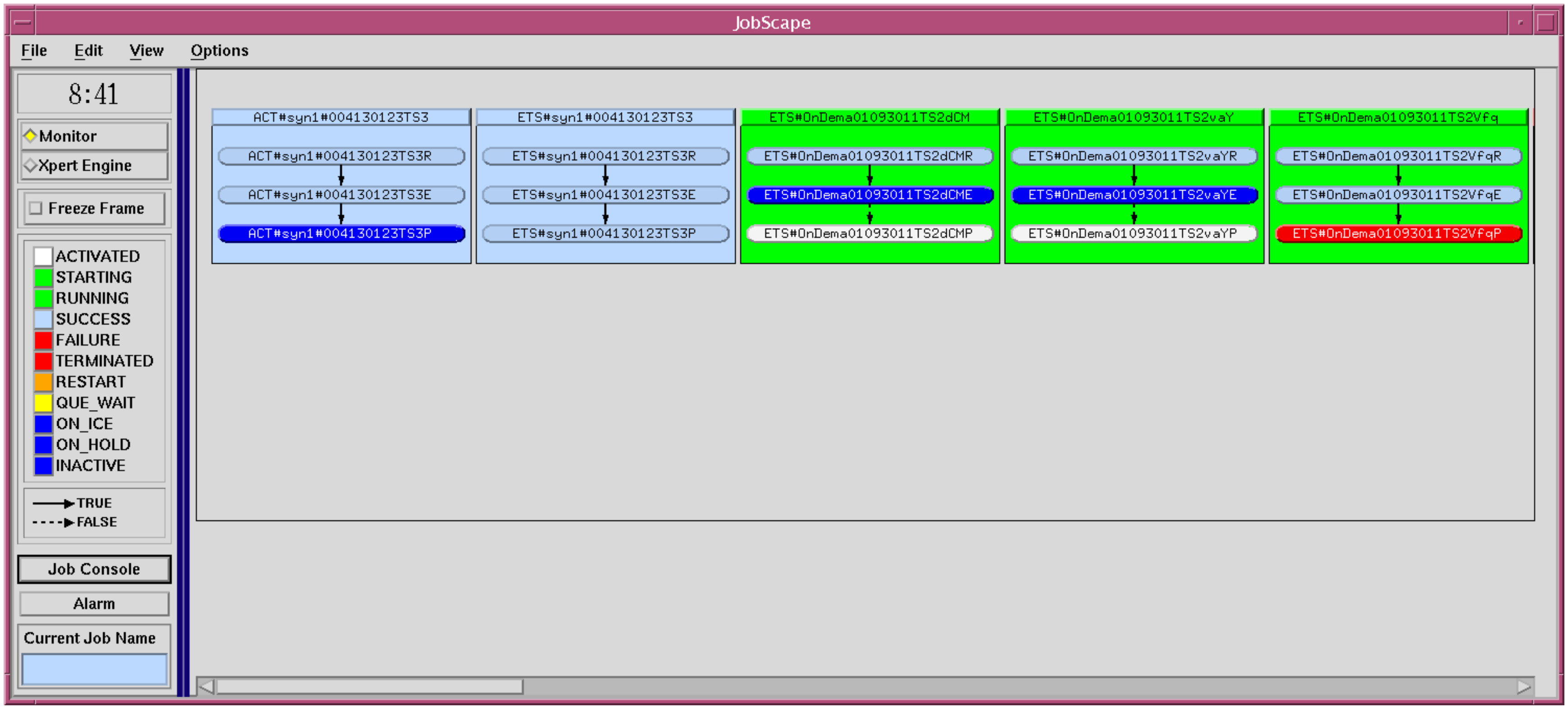
AutoSys GUI Control Panel



TimeScape Interface



JobScape Interface



JobScape Runtime Options Dialogue Box

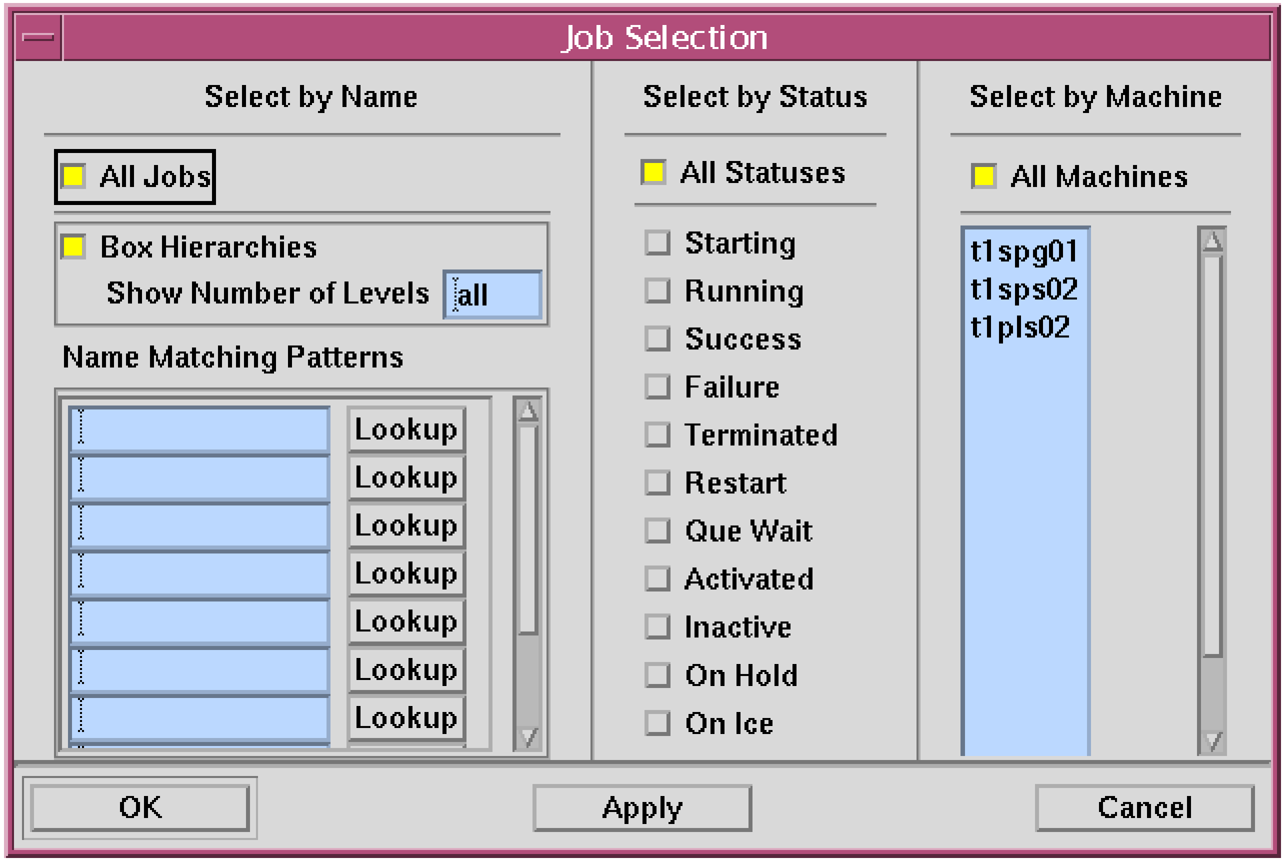
Configuring AutoSys (Cont.): Selecting Jobs to be Displayed

- **Jobs can be selected on the basis of the following criteria:**
  - **Job Name**
  - **Job Status**
  - **Machine**
- **Default values**
  - **All Jobs**
  - **All Statuses**
  - **All Machines**

Configuring AutoSys (Cont.): Selecting Jobs to be Displayed

- **Procedure**
  - **Select either TimeScape or JobScape**
  - **Select View → Select Jobs to Display**
  - **For the Select by Name option select all jobs by clicking on the All Jobs button; to select jobs by pattern matching, enter the appropriate characters/wild cards**
  - **To select all job statuses click on the All Statuses button; to select jobs by status click on the appropriate Select by Status toggle buttons**
  - **To select jobs on all machines click on the All Machines button; to select jobs by machine click on the applicable machine name(s) in the Select by Machine area**
  - **Select OK (or Apply)**

Job Selection Dialogue Box



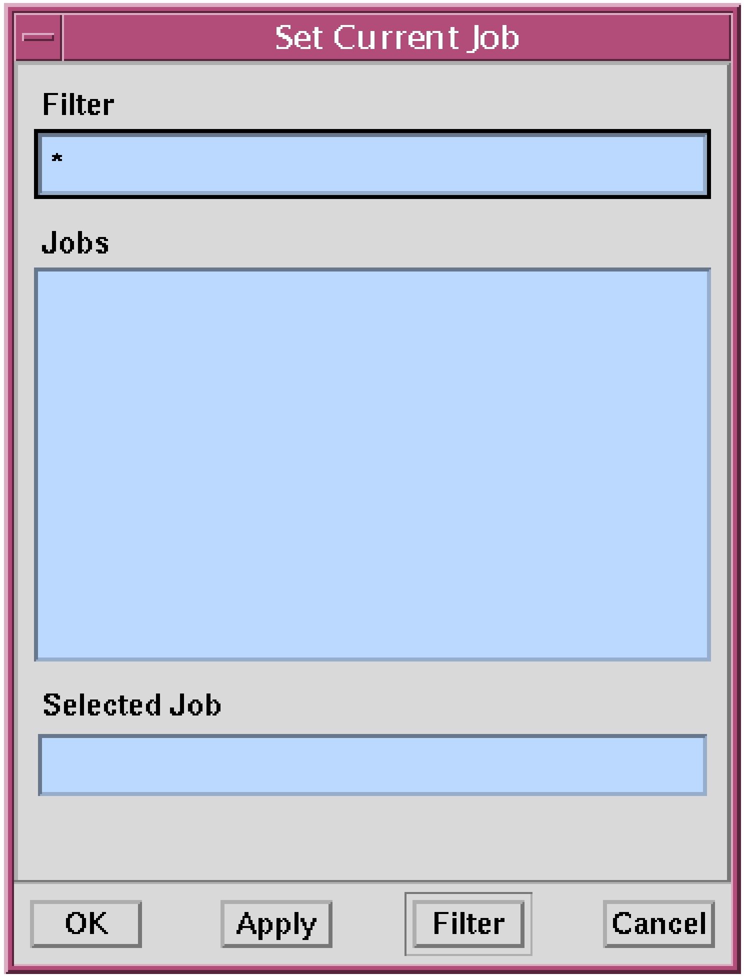
Configuring AutoSys (Cont.): Setting the Current Job

- **Setting the current job causes the job name to be displayed in the Current Job Name field in the Control Region of the display**
  - **Subsequently clicking on the Job Console button on the display causes the Job Activity Console GUI (also known as the Ops Console GUI) to be displayed with information concerning the current job**
- **There are two methods for setting the current job:**
  - **Click on the name of a job displayed on an AutoSys display**
  - **Set the current job using the pull-down menu**

Configuring AutoSys (Cont.): Setting the Current Job

- **Procedure**
  - **Select either TimeScape or JobScape**
  - **Select View → Set Current Job**
  - **Type the *jobname* or a portion of the *jobname* in the Filter field**
    - **The asterisk (*) wildcard character can be used for entering a partial job name**
  - **Click on the Filter button**
  - **Select (highlight) the job to be designated the "current job" from the jobs listed in the Jobs field**
  - **Select OK (or Apply)**

Set Current Job Dialogue Box



Monitoring/Controlling Job Processing

- **There are three primary tools for monitoring and controlling job processing:**
  - **JobScape**
  - **TimeScape**
  - **AutoSys Job Activity Console GUI (also known as also known as the Ops Console GUI)**

Monitoring/Controlling Job Processing (Cont.)

- **JobScape GUI**
  - presents a Pert-like graphical view of job processing from a logical (or job dependency) point of view
  - depicts all job types; i.e., command jobs, box jobs, and file-watcher jobs
  - depicts the nesting of jobs within boxes and the dependencies between jobs
  - can be used for monitoring and controlling job flow in real-time
  - allows the Production Monitor to identify potential problems before they become actual problems, put problem jobs on hold in favor of letting good jobs run, restart jobs after correcting problems with them, etc.

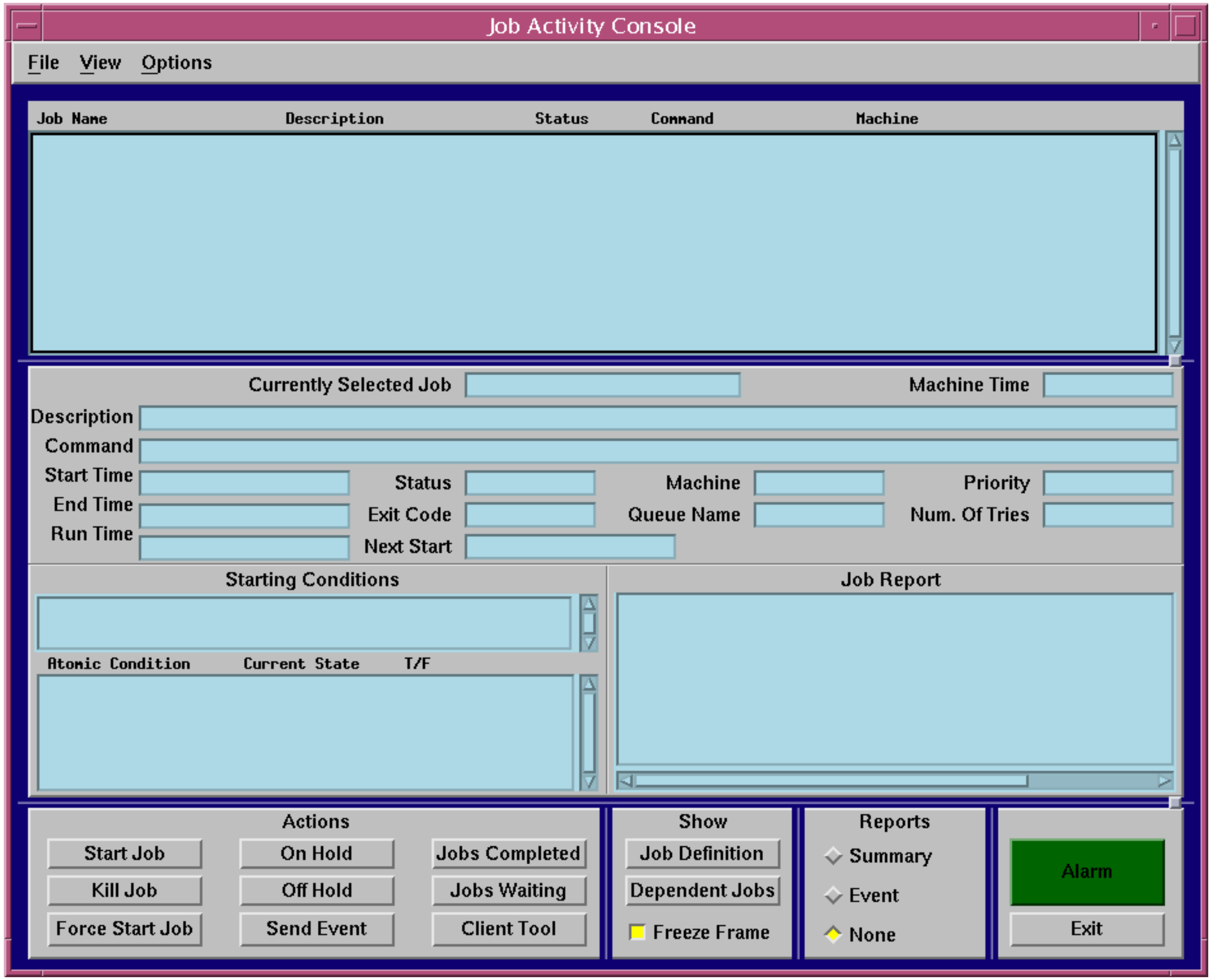
Monitoring/Controlling Job Processing (Cont.)

- **TimeScape GUI**
  - presents a Gantt-like graphical view of a job processing from a temporal (time-related) point of view
  - depicts both command jobs and box jobs
  - depicts the nesting of jobs within boxes and the duration of time it will take for jobs to complete
  - is used for monitoring job flow in real-time

Monitoring/Controlling Job Processing (Cont.)

- Job Activity Console GUI (Ops Console)
  – is a text-based interface for monitoring jobs that have been defined for AutoSys
  – displays information on the job's start time (and date), end time (and date), run time, status, exit code (if completed), host, priority, and other attributes
  – provides a means of evaluating job starting conditions, which can be useful in determining what "upstream" job may be preventing the currently selected job from running
  – provides summary and event reports that can be used in identifying problems with processing a particular job

Job Activity Console (OPS Console)



Monitoring/Controlling Job Processing (Cont.)

- **AutoSys defines job status in the terms listed in the table**
  - **different states are color-coded on JobScape and TimeScape displays**

Monitoring/Controlling Job Processing (Cont.)

- **Job States**
  - Activated (white)
  - Starting (green)
  - Running (green)
  - Success (light blue)
  - Failure (red)
  - Terminated (red)
  - Restart (orange)
  - Que_Wait (yellow)
  - On_Ice (dark blue)
  - On_Hold (dark blue)
  - Inactive (dark blue)

Monitoring/Controlling Job Processing (Cont.)

- **Production Monitor may need to modify a particular job in any of the following ways:**
  - **Start the job**
  - **Kill the job**
  - **Force the job to start**
  - **Place the job on hold**
  - **Take the job off hold**
- **Three methods for making those types of modifications to a particular job:**
  - **Buttons in the Actions region of the Job Activity Console (Ops Console)**
  - **Menu accessed by clicking the right mouse button on the relevant job name on JobScape or TimeScape**
  - **AutoSys Send Event GUI**

Monitoring/Controlling Job Processing (Cont.)

- In AutoSys terms a control action such as starting or killing a job is accomplished by sending an “event” to the job
  - An event is basically a message
  - For example:
    - Clicking on the Start Job button on the AutoSys Job Activity Console begins the process by which AutoSys sends a “start” message to the Currently Selected Job
- In addition to modifications to job status, the buttons in the Actions region of the Job Activity Console (Ops Console) allow the Production Monitor to generate reports:
  - Jobs Completed
  - Jobs Waiting

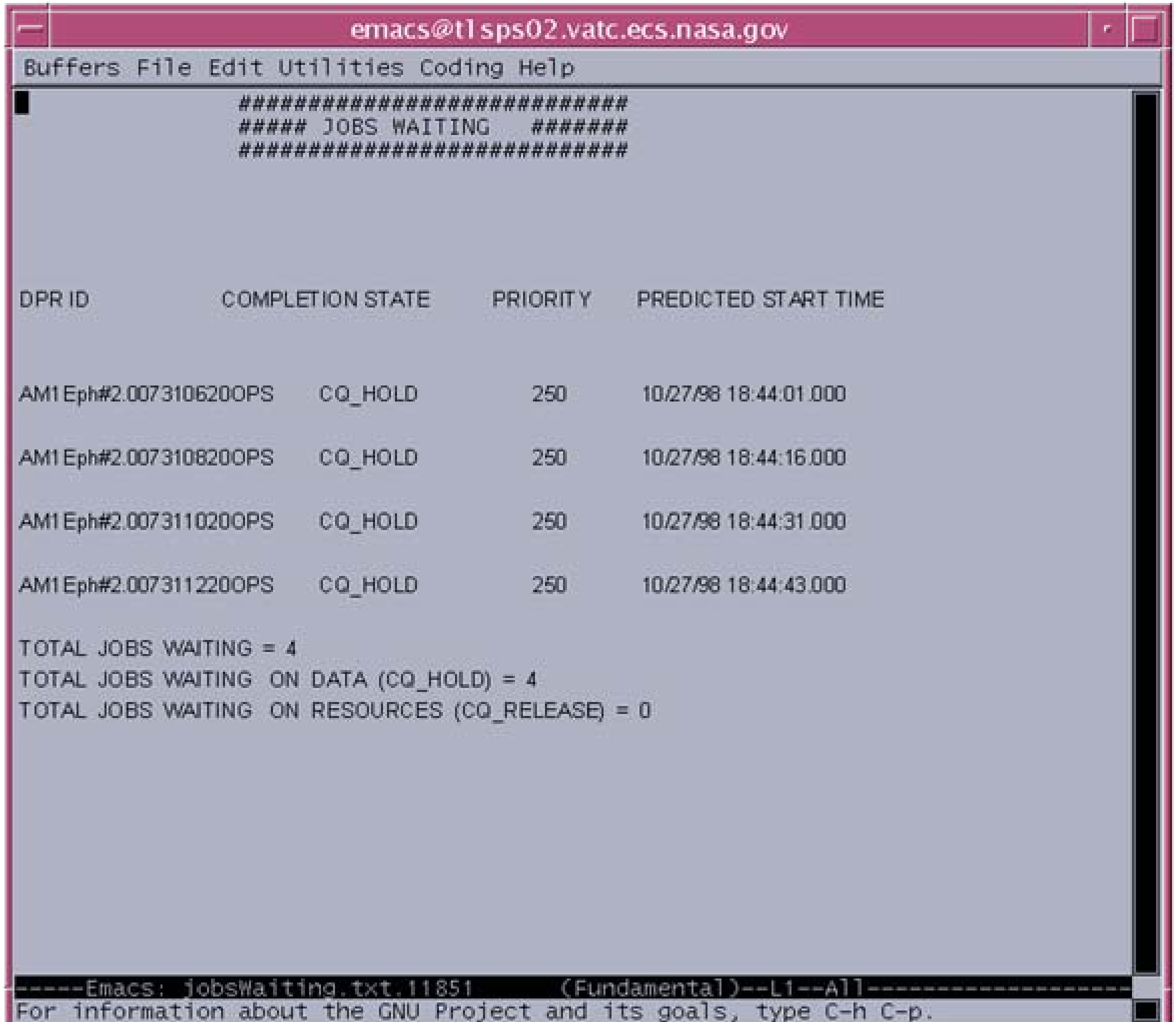
Jobs Completed Report

```
emacs@t1sps02.vatc.ecs.nasa.gov
Buffers File Edit Utilities Coding Help
                ############################
                ##### JOBS COMPLETED #######
                ############################

DPR ID                  COMPLETION STATE    PRIORITY  DATA START TIME     AUTOSYS

MISRLIKE#s03050000TS    SUCCESS               2750   12/03/93 00:00:00.000   VAT
BTS#OnDema01093011TS    SUCCESS               120   04/01/95 04:30:11.000   VAT
ACT#syn1#004130123TS    SUCCESS               2750   07/04/97 09:01:23.000   VAT
BTS#syn1#004130123TS    SUCCESS               2750   07/04/97 09:01:23.000   VAT
FddAtt#30112312200TS    SUCCESS               2750   12/31/99 17:00:00.000   VAT
AM1Eph#30101010000TS    CANCELED              2750   12/31/99 19:00:00.000   VAT
FddAtt#30101010000TS    SUCCESS               2750   12/31/99 19:00:00.000   VAT
MoPGE01#sy01010000TS    CANCELED              2750   12/31/99 19:00:00.000   VAT
MoPGE02#sy01000500TS    CANCELED              2750   12/31/99 19:05:00.000   VAT
MoPGE03#sy01000500TS    CANCELED              2750   12/31/99 19:05:00.000   VAT
MoPGE02#sy01001000TS    CANCELED              2750   12/31/99 19:10:00.000   VAT
MoPGE03#sy01001000TS    CANCELED              2750   12/31/99 19:10:00.000   VAT
MoPGE02#sy01001500TS    CANCELED              2750   12/31/99 19:15:00.000   VAT
MoPGE03#sy01001500TS    CANCELED              2750   12/31/99 19:15:00.000   VAT
MoPGE02#sy01002000TS    CANCELED              2750   12/31/99 19:20:00.000   VAT
-----Emacs: jobsCompleted.txt.11451      (Fundamental)--L1--Top----------------
For information about the GNU Project and its goals, type C-h C-p.
```

Jobs Waiting Report



emacs@t1sps02.vatc.ecs.nasa.gov

Buffers File Edit Utilities Coding Help

```
##############################
##### JOBS WAITING   #######
##############################
```

DPR ID	COMPLETION STATE	PRIORITY	PREDICTED START TIME
AM1Eph#2.007310620OPS	CQ_HOLD	250	10/27/98 18:44:01.000
AM1Eph#2.007310820OPS	CQ_HOLD	250	10/27/98 18:44:16.000
AM1Eph#2.007311020OPS	CQ_HOLD	250	10/27/98 18:44:31.000
AM1Eph#2.007311220OPS	CQ_HOLD	250	10/27/98 18:44:43.000

TOTAL JOBS WAITING = 4
TOTAL JOBS WAITING ON DATA (CQ_HOLD) = 4
TOTAL JOBS WAITING ON RESOURCES (CQ_RELEASE) = 0

-----Emacs: jobsWaiting.txt.11851 (Fundamental)--L1--All-----------------

For information about the GNU Project and its goals, type C-h C-p.

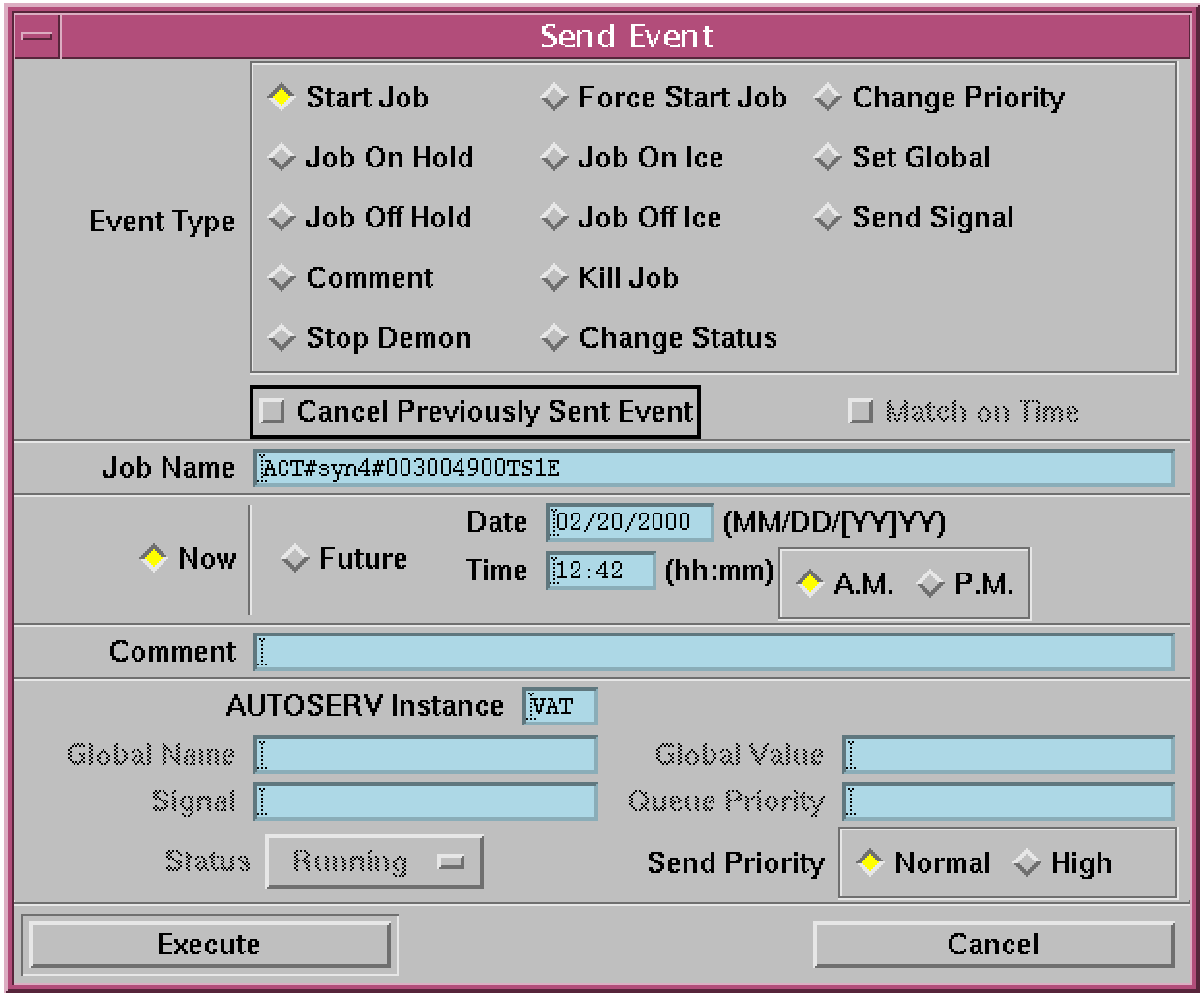
Monitoring/Controlling Job Processing (Cont.)

- The menu accessed using the right mouse button on either JobScape or TimeScape allows the Production Monitor to initiate either of the following actions (in addition to the previously mentioned modifications to job status):
  – Put the job on ice
  – Take the job off ice

Monitoring/Controlling Job Processing (Cont.)

- **Send Event GUI allows the Production Monitor to initiate any of the following actions:**
  - Start the job
  - Kill the job
  - Force the job to start
  - Place the job on hold
  - Take the job off hold
  - Change the job's status
  - Change the job's priority
  - Put the job on ice
  - Take the job off ice
  - Stop the daemon (stop the Event Processor in an emergency)
  - Set a global value
  - Send a signal concerning the job
  - Make a comment (for example, why a job start was forced)

Send Event GUI



Monitoring/Controlling Job Processing (Cont.)

- **Guidelines for Reporting Unsuccessful Completion of On-Demand Jobs**
  - **Under any of the following circumstances involving an on-demand job notify User Services of the problem in accordance with the applicable local policy**
    - **Job is killed**
    - **Job terminates and cannot be restarted**
    - **A FAILPGE granule is created**
  - **The DAAC is obliged to send an e-mail message to the requester of an unsuccessful on-demand job to explain why the request cannot be fulfilled**

Monitoring/Controlling Job Processing (Cont.)

- **Guideline for Putting Jobs "On Ice" or "On Hold"**
  - **Ensure that the job to be put either "on hold" or "on ice" is not already in a "starting" or "running" state**
    - **A job that is either "starting" or "running" cannot be put "on hold" or "on ice"**

Monitoring/Controlling Job Processing (Cont.)

- **Guidelines for Force-Starting Jobs**
  - **Force-start command jobs (e.g., preprocessing or preprocessing) only**
  - **Do not attempt to force-start a box job**
    - **The software does not support box job force-starts**
    - **Force-starting a box job can cause the PDPS database to get out of sync and prevent the DPR (and possibly other DPRs) from running successfully**
    - **If a box job were force-started, the allocation portion of the preprocessing job job would run again and might choose a different science processor than was chosen the previous time the job ran, which could cause failure of the job**
    - **Box job force-starts lack the code needed to check the state of the box and perform the database cleanup activities necessary for starting over**

Monitoring/Controlling Job Processing (Cont.)

- **Guidelines for Force-Starting Jobs (Cont.)**
  - Ensure that the GUI has refreshed and the job to be force-started is not already running before trying to force-start a job
    - If a job is already running, it should not be force-started
    - It should not be possible to force-start jobs that are already running
  - If any command job other than execution fails, force-start the job that failed only
  - Do not force start any preceding or succeeding jobs in the box

Monitoring/Controlling Job Processing (Cont.)

- **Guidelines for Force-Starting Jobs (Cont.)**
  - If execution fails, it is not safe to restart it unless the post-processing job had been put on hold and the failure was detected before post-processing started running
  - If execution fails and the failure was not detected before post-processing started running, the DPR must run to completion as a failed PGE and the DPR must be deleted and recreated
  - In any case the Production Monitor may implement certain changes of job status only when the Production Monitor "owns" the job affected by the modification

Monitoring/Controlling Job Processing (Cont.)

- Procedure (Monitoring/Controlling Job Processing)
  - Click on AutoSys GUI Control Panel buttons to display interfaces (as necessary)
    - JobScape
    - TimeScape
    - Job Activity Console (Ops Console)
  - Configure runtime options for JobScape/TimeScape as necessary
  - If necessary, select jobs to be displayed on JobScape/TimeScape
  - Observe information displayed on JobScape/TimeScape/Job Activity Console
  - Perform subordinate procedures as necessary
  - Repeat preceding steps as necessary to monitor/control jobs

Monitoring/Controlling Job Processing (Cont.)

Good vs Bad

Looks Good!

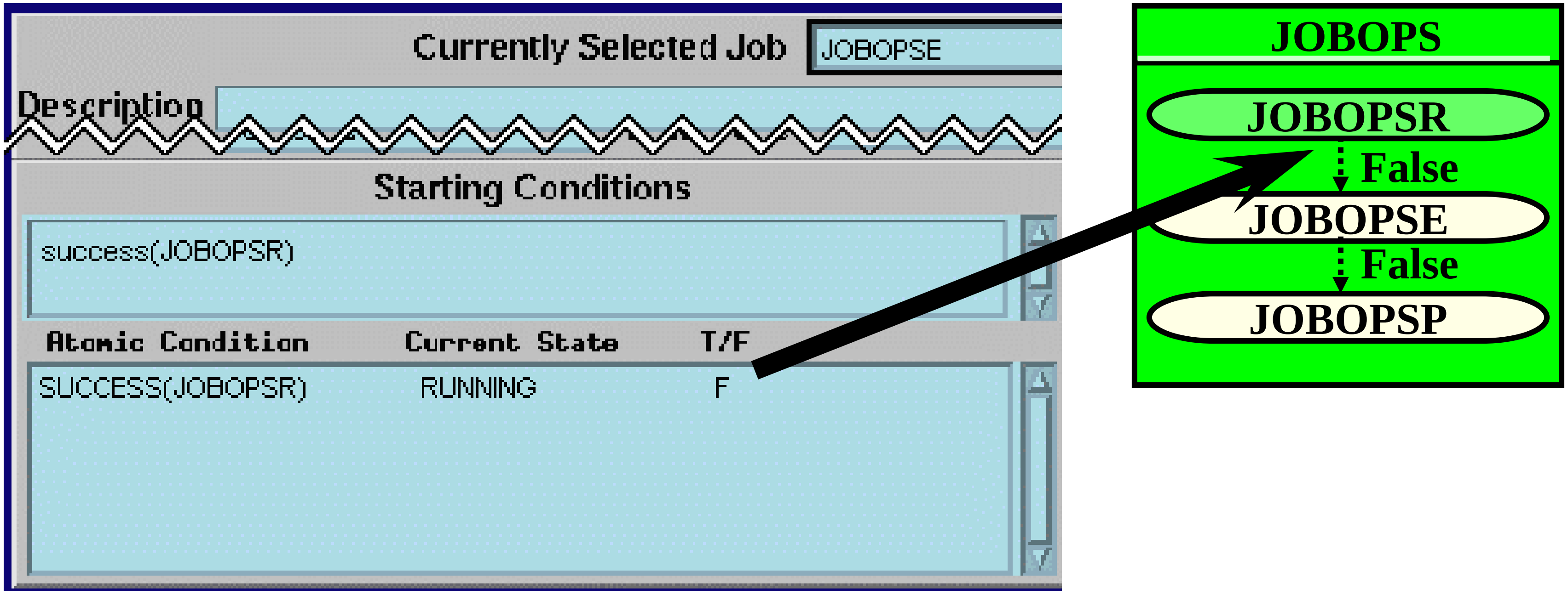
Trouble!

Monitoring/Controlling Job Processing (Cont.)

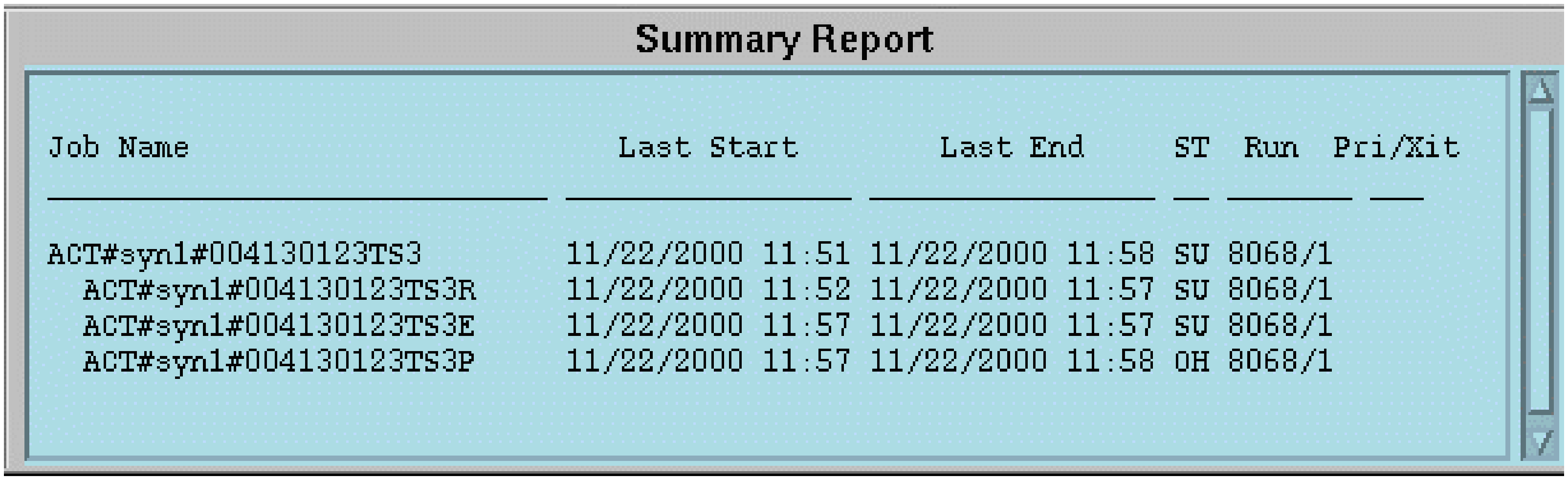
Nuke'm

Job Console

JobScape GUI



Example of A Summary Report on the Job Activity Console



Summary Report

Job Name	Last Start	Last End	ST	Run	Pri/Xit
ACT#syn1#004130123TS3	11/22/2000 11:51	11/22/2000 11:58	SU	8068/1	
ACT#syn1#004130123TS3R	11/22/2000 11:52	11/22/2000 11:57	SU	8068/1	
ACT#syn1#004130123TS3E	11/22/2000 11:57	11/22/2000 11:57	SU	8068/1	
ACT#syn1#004130123TS3P	11/22/2000 11:57	11/22/2000 11:58	OH	8068/1	

Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Determining the Descendants of a Job)**
  - **Select the job by placing the mouse cursor on the job and clicking with the left mouse button**
  - **Place the mouse cursor on the job and click and hold the right mouse button**
  - **If applicable, select (highlight) Show Children from the pop-up menu (release the right mouse button)**
  - **If applicable, select Show All Descendants from the pop-up menu**
  - **If applicable, select Hide All Descendants from the pop-up menu**

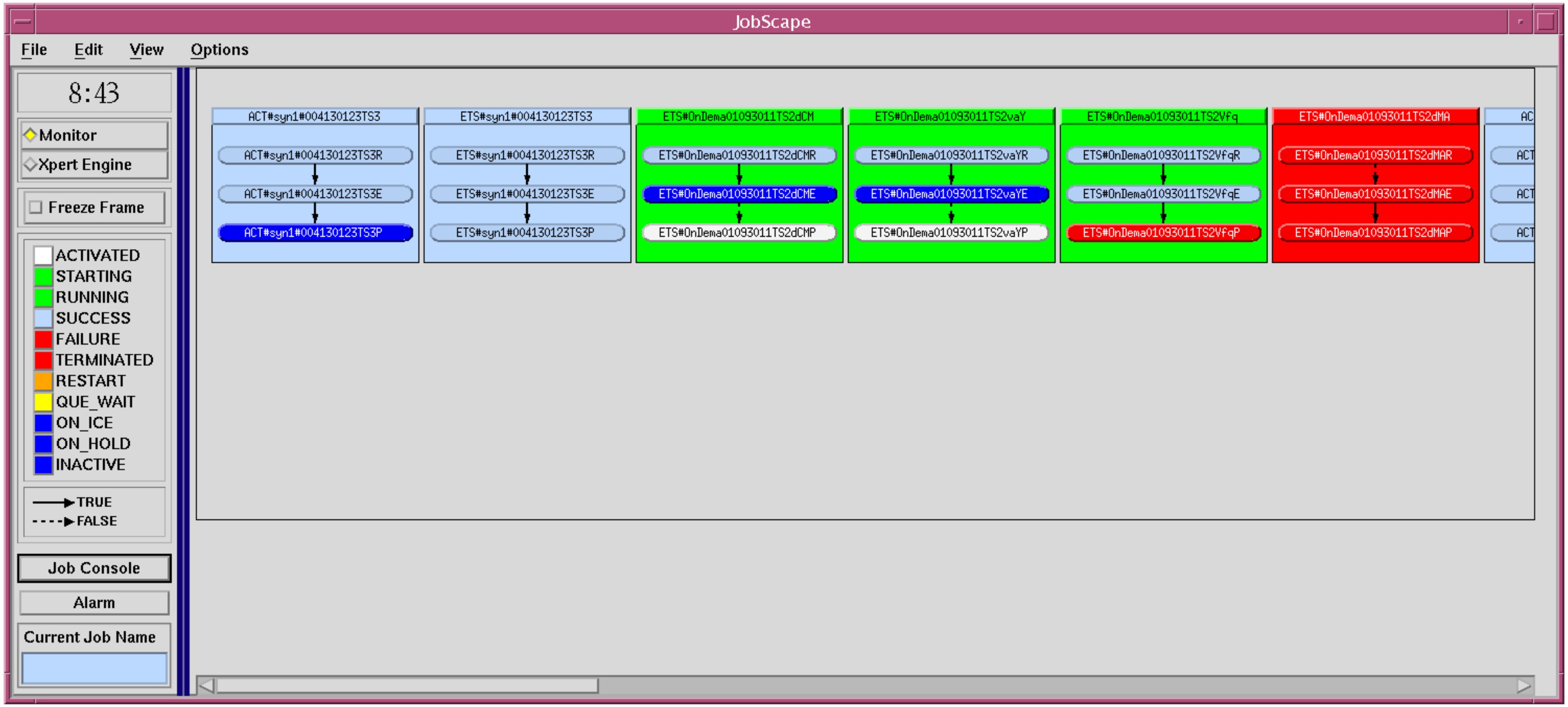
Monitoring/Controlling Job Processing (Cont.)

- **Changing the JobScape View Using the Pull-Down Menu**
  - – **Changing the view affects the level of detail displayed in the View Region of the GUI**
  - – **The view can be changed in two ways**
    - **Simply clicking with the right mouse button on the name of a job displayed on a display and selecting the desired option from the pop-up menu**
    - **Using the View pull-down menu (has some additional options)**

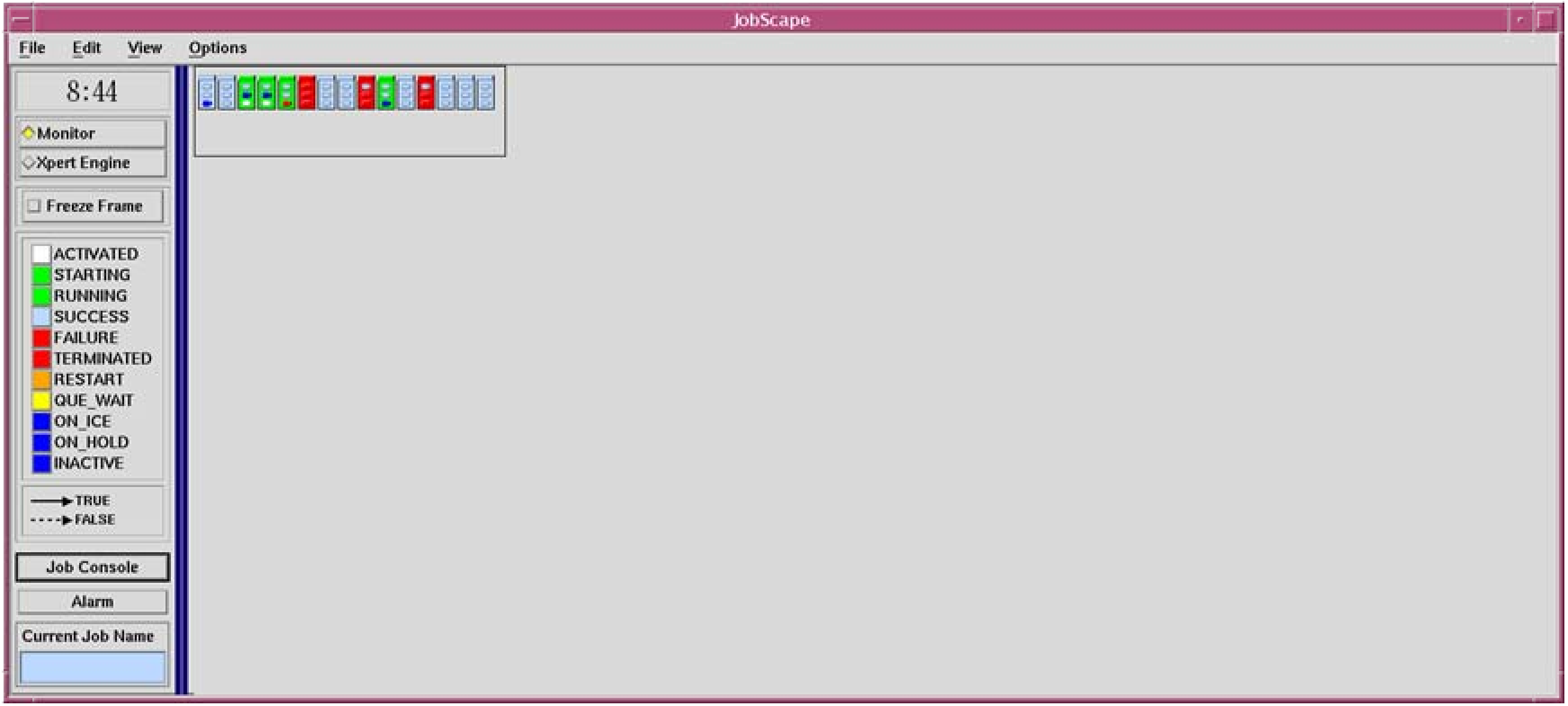
Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Changing the JobScape View Using the Pull-Down Menu)**
  - **Select View → Set View from the pull-down menu**
  - **Select the desired option from the pull-down menu**
    - **Normal Text View**
    - **Small Text View**
    - **No Text View**
    - **Show Arrows**
    - **Hide Arrows**
    - **View by Id**
  - **Select View → Set Display Levels**
  - **Select the desired option from the pull-down menu**
    - **All is the default type of view**

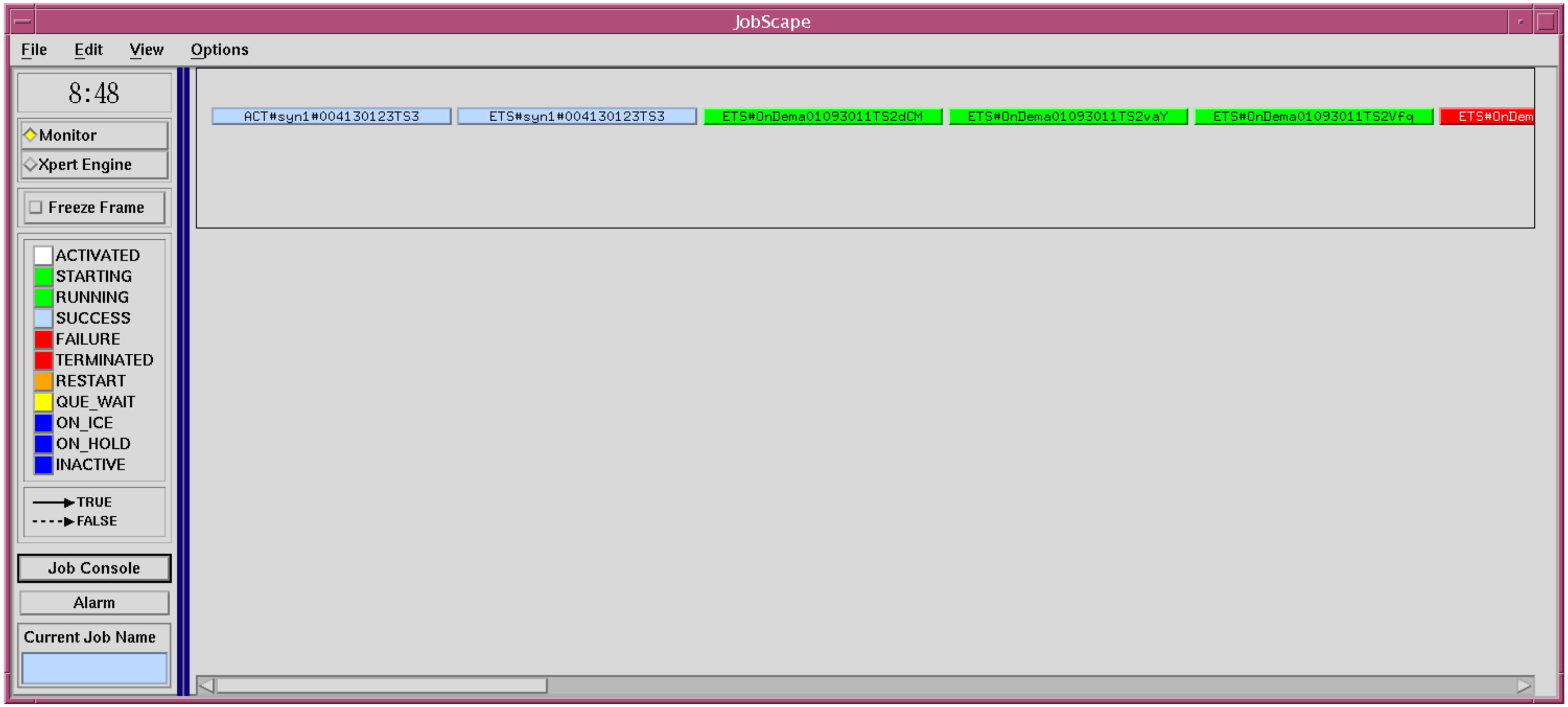
JobScape GUI “Small Text” View



JobScape GUI "No Text" View



JobScape GUI "One Level" View



Monitoring/Controlling Job Processing (Cont.)

- Reviewing Alarms
  - Alarms indicate problems with job processing
    - failure of job processing
    - database problem
    - communication problem
    - hardware or software failure
    - some other error in the data processing system
  - Production Monitor reviews alarms using the AutoSys Alarm Manager
    - view alarms as they arrive
    - provide a response to an alarm
    - change alarm status

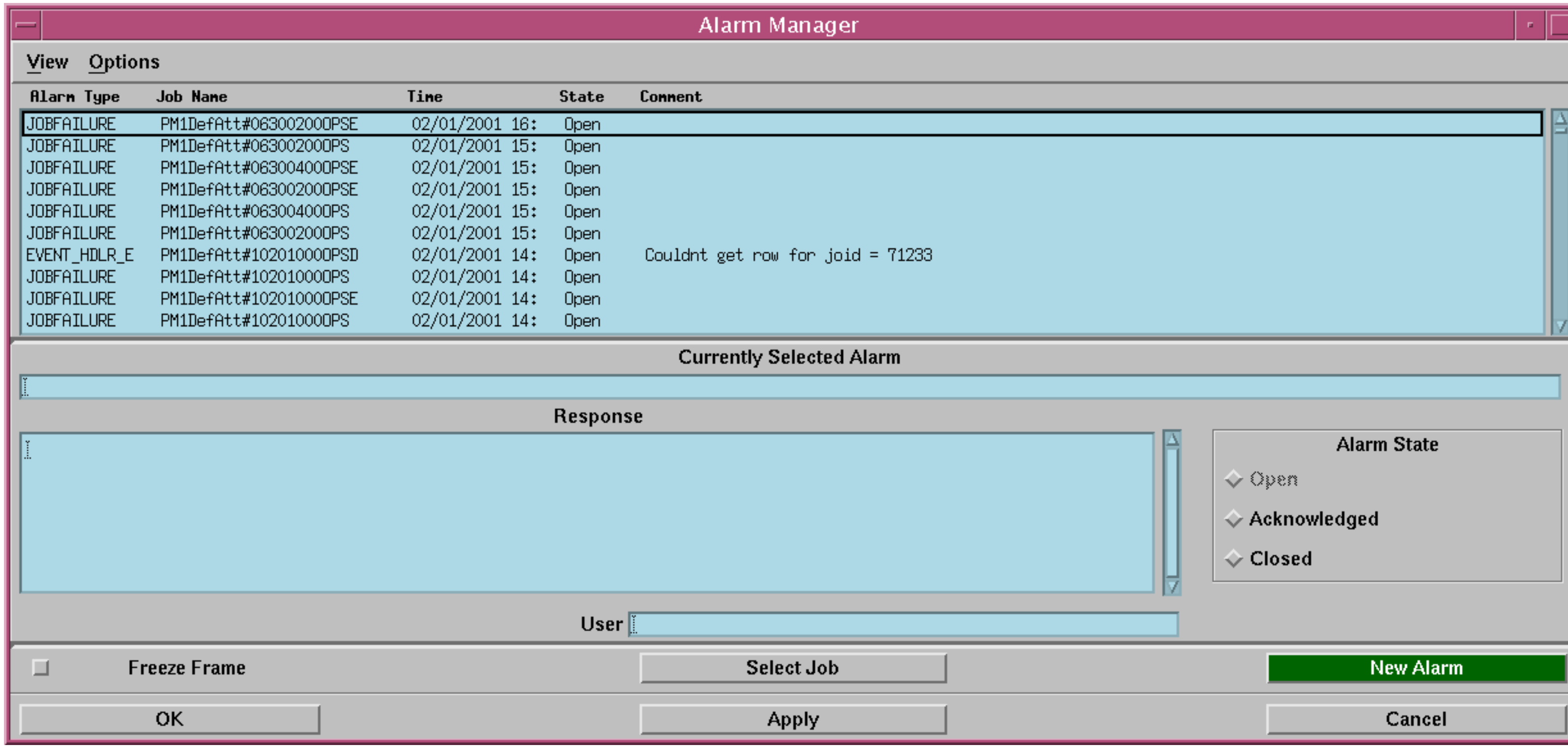
Monitoring/Controlling Job Processing (Cont.)

- **Reviewing Alarms (Cont.)**
  – Production Monitor can configure the Alarm Manager to display certain types of alarms only
    - type of alarm
    - alarm state
    - time of the alarm
  – For descriptions of AutoSys alarms refer to the AutoSys® Reference Guide for UNIX
    - The AutoSys® Reference Guide for UNIX, the AutoSys® User Guide for UNIX, and the AutoSys®/Xpert User Guide for UNIX can be downloaded from the Computer Associates Technical Support website but require an account and login
    - Contact the DAAC COTS software representative for assistance

Monitoring/Controlling Job Processing (Cont.)

- Procedure (Reviewing Alarms)
  – Click on the Alarm button to display the Alarm Manager GUI
  – Configure alarm selection as described in the procedure for Configuring Alarm Selection
  – Click on an alarm in the Alarm List
  – Click the Response edit box and type in a response, if desired
  – Update the Alarm State by clicking on the appropriate radio button
  – Click on the Apply button
  – Repeat steps as necessary to review and update multiple alarms
  – Click on the OK button to dismiss the Alarm Manager GUI

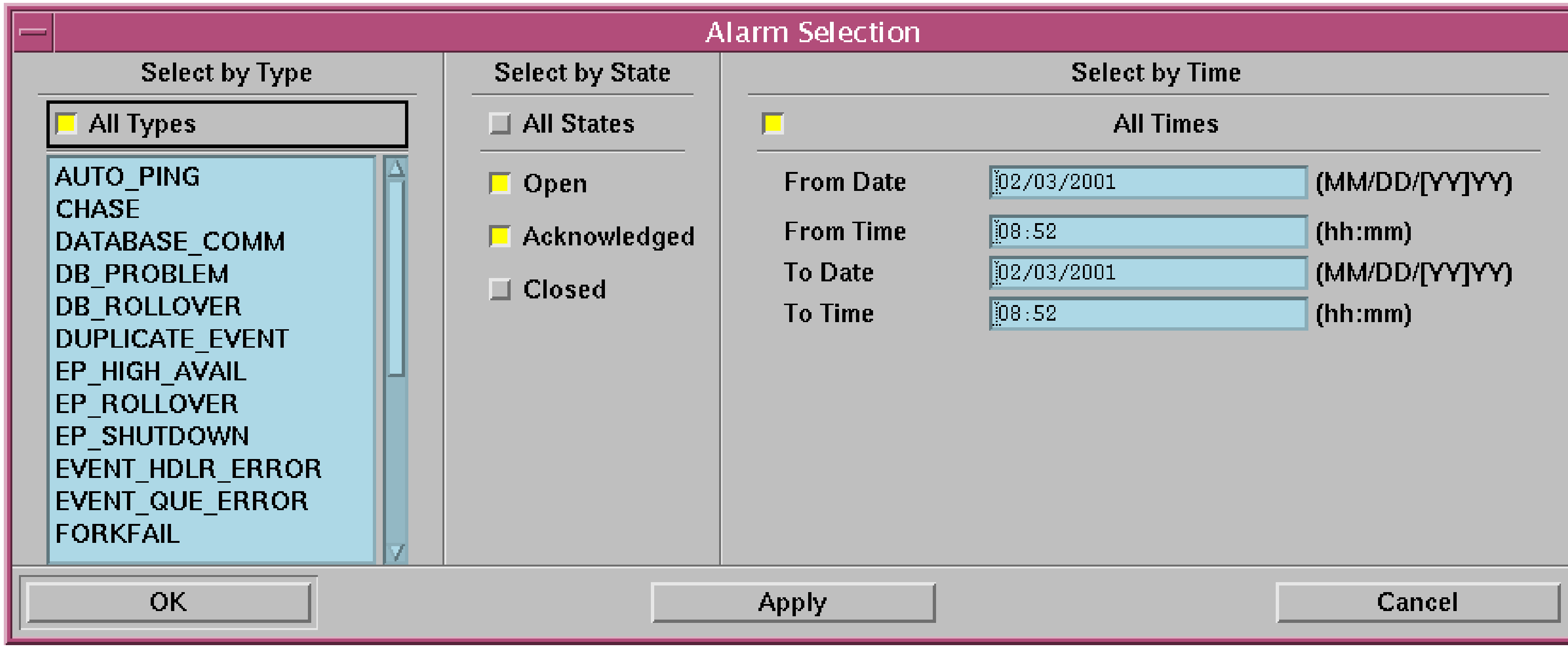
Alarm Manager GUI



Monitoring/Controlling Job Processing (Cont.)

- Procedure (Configuring Alarm Selection)
  - Select View → Select Alarms from the pull-down menu to display the Alarm Selection GUI
  - Click on the desired alarm(s) in the Select by Type alarm list; to select all types of alarms, click on the All Types button
  - Click on the All States button to select all alarm states; to select alarms by state click on whichever of the Select by State toggle buttons properly describe(s) the state(s) to be selected
  - Click on the All Times button to select all times; to select alarms by time type the starting date/time and ending date/time in the applicable fields
  - Apply selections

Alarm Selection GUI



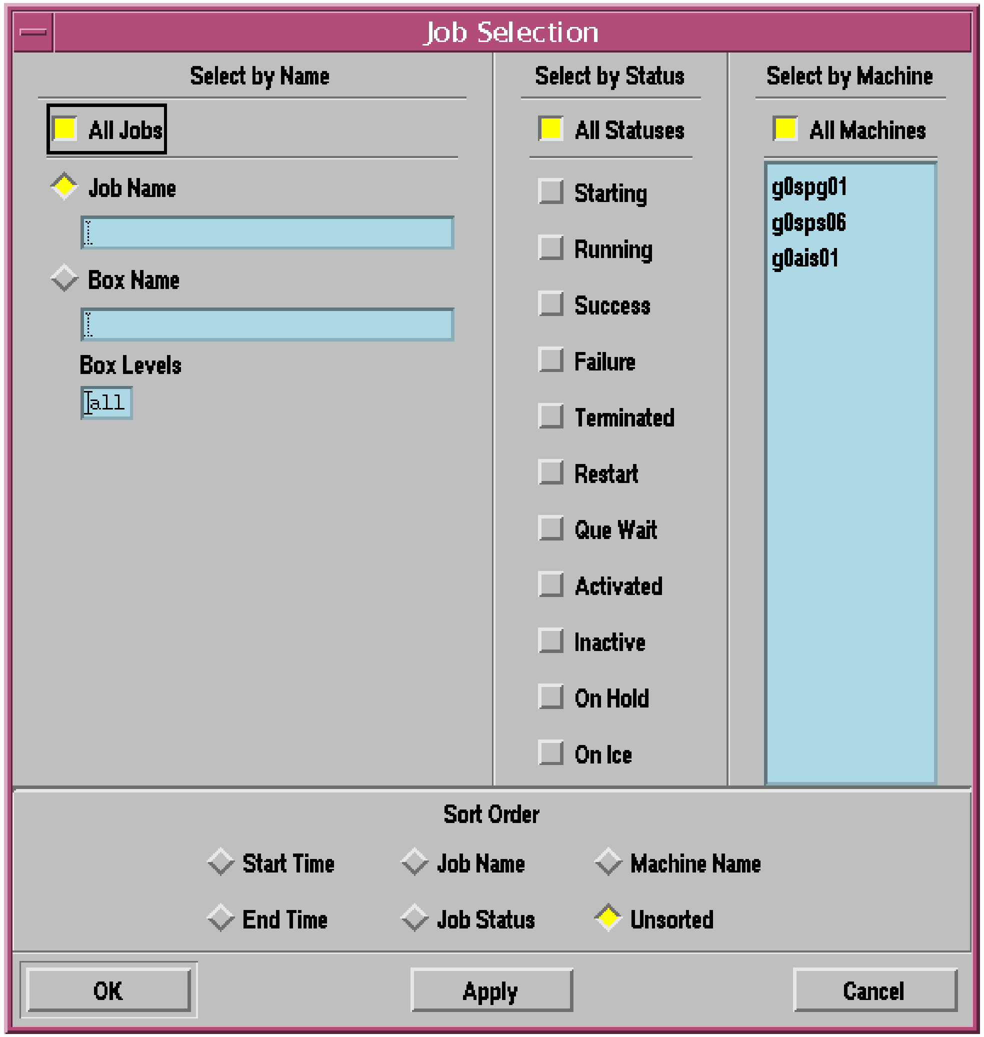
Monitoring/Controlling Job Processing (Cont.)

- **Specifying Job Selection Criteria**
  - Production Monitor reviews job activities using the AutoSys Job Activity Console
  - AutoSys Job Selection GUI is used for...
    - specifying (filtering) jobs the Production Monitor will review
    - setting the criteria for displaying jobs by name, status and/or machine

Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Specifying Job Selection Criteria)**
  - Click on the Ops Console button on the AutoSys GUI Control Panel
  - Choose View → Select Jobs from the pull-down menu to display the Job Selection GUI
  - For the Select by Name option select all jobs by clicking on the All Jobs button
  - To select all job statuses click on the All Statuses button; to select jobs by status click on the appropriate Select by Status toggle buttons
  - To select all machines click on the All Machines button; to select individual machines click on the machines in the list in the Select by Machine area
  - Click on the desired order in the Sort Order area
  - Apply the selections

Job Selection GUI



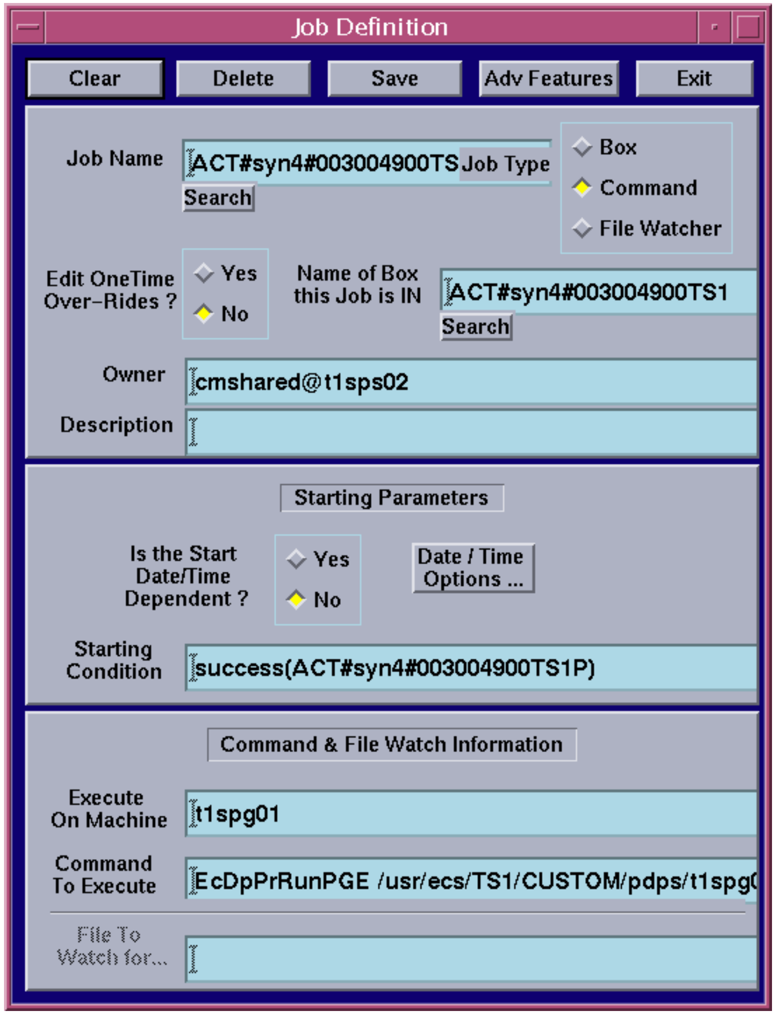
Monitoring/Controlling Job Processing (Cont.)

- **Determining the Ownership of an AutoSys Job**
  - **AutoSys is very much ownership-aware**
    - **Only the "owner" of a job has "edit" privileges and can make changes to the status of an owned job**
  - **AutoSys recognizes ownership in terms of two factors:**
    - **UserID**
    - **Machine where the operator (user) logged in**
  - **Example:**
    - **cmshared@g0sps06 identifies the Production Monitor who logged in as "cmshared" at g0sps06**
    - **Any operator who logs in as "cmshared" at another machine (e.g., g0pls01) would not be able to change the status of a job "owned" by cmshared@g0sps06**
    - **To have any real effect on a job first it is necessary to log in as the job's owner and launch AutoSys GUIs**

Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Determining the Ownership of a Job)**
  - Click on the JobScape button (or TimeScape button) on the AutoSys GUI Control Panel
  - Place the mouse cursor on the relevant job and click and hold the right mouse button
  - Select (highlight) Job Definition from the pop-up menu (release the right mouse button)
  - Review the entry in the Owner field of the Job Definition GUI
  - To exit from the Job Definition GUI, click on the Exit button

Job Definition GUI



Job Definition GUI
Job Security MESSAGE Window

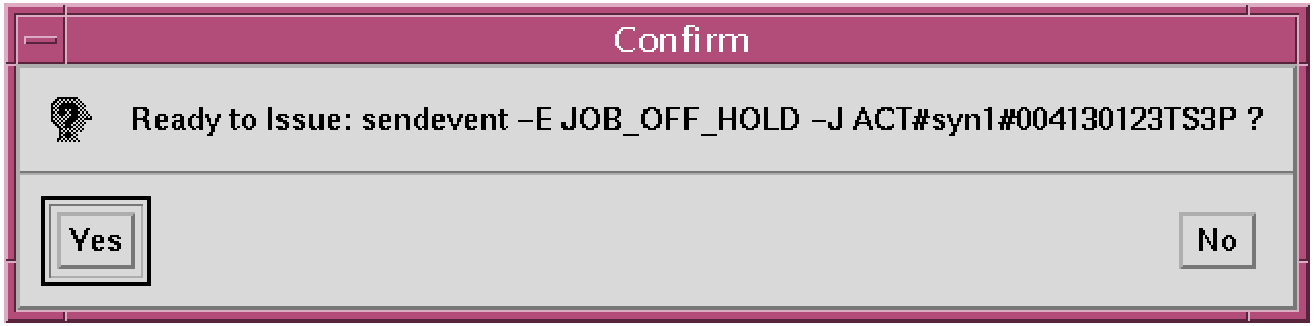
Monitoring/Controlling Job Processing (Cont.)

- **Sending an Event to a Job**
  - **As previously mentioned there are three methods for making certain types of modifications (e.g., start or kill) to a particular job**
    - **Menu accessed by clicking the right mouse button on the relevant job name on either the JobScape or TimeScape GUI**
    - **Buttons in the Actions region of the Job Activity Console (Ops Console)**
    - **AutoSys Send Event GUI**

Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Sending an Event to a Job from JobScape or TimeScape)**
  - Select either JobScape or TimeScape from the AutoSys GUI Control Panel
  - Place the mouse cursor on the relevant job and click and hold the right mouse button
  - Select the event (e.g., Force Start Job) to be sent to the job from the pop-up menu
    - If there is no option corresponding to the desired action, modify job status from either the Job Activity Console or the Send Event GUI
  - Select Yes to confirm sending the event

Confirmation Dialogue Box



Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Sending an Event to a Job from the Job Activity Console)**
  - **Select jobs for display on the Job Activity Console (Ops Console)**
  - **Verify that the job with the status to be modified is listed in the Currently Selected Job field of the Job Activity Console**
  - **Click on the button corresponding to the desired action to be taken with respect to the selected job**
    - **If there is no button corresponding to the desired action, modify job status using the Send Event GUI**
  - **Click on Yes to send the event to the job**

Monitoring/Controlling Job Processing (Cont.)

- **Sending an Event to a Job from the Send Event GUI**

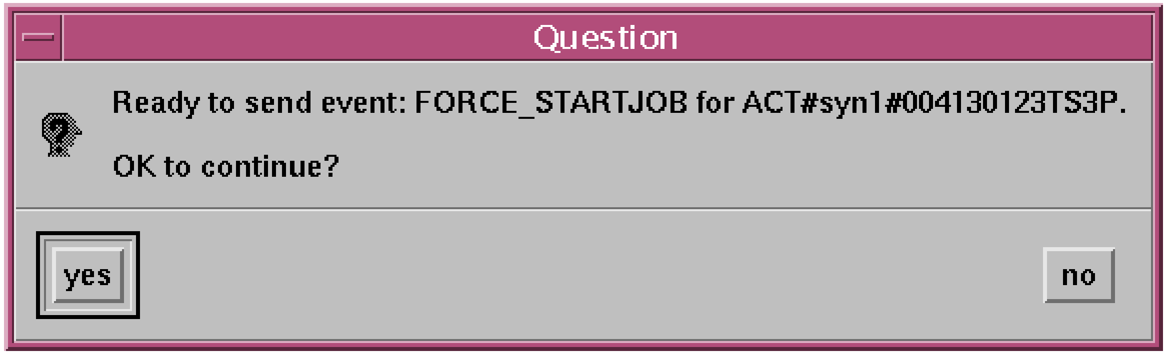
CAUTION

Once an event has been sent from the Send Event dialog, it may not be possible to cancel or modify it.

Monitoring/Controlling Job Processing (Cont.)

- Procedure (Sending an Event to a Job from the Send Event GUI)
  – Click on the Ops Console button and list jobs
  – In the Job List region of the Job Activity Console click on the job row corresponding to the job
  – Click on the Send Event button
  – Select the Event Type
  – Enter the desired date and time when the job status is to be modified
  – Change the Queue Priority entry if Change Priority was selected
  – Select Status to send if Change Status was selected
  – Select the Send Priority
  – Click on the Execute button

Send Event GUI: Confirmation Dialogue Box



Monitoring/Controlling Job Processing (Cont.)

- Procedure (Canceling a Sent Event)
  - Click on the Send Event button in the Actions Region of the Job Activity Console
  - Click on Event Type to select the type of event that was sent to the job and is to be canceled
  - Click on the Cancel Previously Sent Event button
  - Verify that the correct job is listed in the Job Name field of the Send Event GUI
  - Click on the Execute button
  - Click on Yes to confirm canceling the sent event

Monitoring/Controlling Job Processing (Cont.)

- **Performing Job Management Client Functions**
  - **Job Management Client tool is a set of utility programs intended primarily for use by software developers**
  - **Get access to any of the following Job Management Client functions from AutoSys by clicking on the Client Tool button in the Actions region of the Job Activity Console**
    - **Create DPR Job**
    - **Release DPR Job**
    - **Cancel DPR Job**
    - **Change DPR ID**
    - **View Job Management DPR Queue**
    - **Create Ground Event Job**
    - **Cancel Ground Event Job**

Monitoring/Controlling Job Processing (Cont.)

- **Performing Job Management Client Functions (Cont.)**
  - **Get access to Job Management Client functions (Cont.)**
    - **Change Max Concurrent Jobs for PGE Limits table**
    - **Cancel Max/Min Dprs for Job Class**
    - **Trigger release of unreleased ready-to-run Dprs**

Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Performing Job Management Client Functions)**
  - Verify that a box job (e.g., a box job with status to be modified) is listed in the Currently Selected Job field of the Job Activity Console (Ops Console)
  - Click on the Client Tool button
  - Enter the number corresponding to the desired function at the “enter an option” prompt
  - Enter responses to Job Management Client prompts

Job Activation User Interface Window

Jobs Activation User Interface

```
MinorVersion = 0
SubSysName = DPS
AppLogLevel = 0
AppLogSize = 200000
DebugLevel = 3
AppStrtNum = 12345
DBHandleList = PDPS
DBName = pdps
DBModeOverride = NONE
DBServer = p0pls02_srvr
DBLibrary = SYBASE_CT
SYBASE = /tools/syb0Cv11.1.1EBF
SYBINTERFACES = /tools/syb0Cv11.1.1EBF/interfaces
Update_Chain_Tables_Script = CUSTOM/bin/DPS/EcDpPrLoadTable.pl

Creating DpPrSchedulerProxy object...



Client Path: 0x196d71c

01/14/02 07:55:11: Client Successfully connected to the server object

AM1Eph#syn032812000PS is an invalid job box

*** Current DprId:NONE Current Mode:TS2 ***

 0) Exit
 1) Create Dpr Job
 2) Release Dpr Job
 3) Cancel Dpr Job
 4) Change Dpr Id
 5) View Job Management Dpr Queue
 6) Create Ground Event Job
 7) Cancel Ground Event Job
 8) Change Max Concurrent Jobs for PGE Limits table
 9) Change Max/Min Dprs for Job Class
 a) Trigger release of unreleased ready-to-run Dprs

enter an option:
```

Reviewing Activity and Job Dependency Reports

- **Reviewing a Job Activity Report**
  - **Production Monitor reviews a job activity report to determine...**
    - **which jobs are currently in the AutoSys queue**
    - **which jobs have been completed**
    - **the completion status of jobs that have been completed**
    - **which jobs are currently running**

Sample Job Activity Report

Terminal

Window Edit Options Help

```
                              5/2001 14:38 01/25/2001 14:38 SU 8508/1

t1sps02:/usr/ecs/TS2/CUSTOM/utilities[62] > autorep -J ETS#OnDema01093011TS2

Job Name                        Last Start         Last End       ST   Run   Pri/Xit
____________________________ ________________ ________________ __ _______ ___

ETS#OnDema01093011TS2        01/25/2001 18:01 -----            RU 8511/1
  ETS#OnDema01093011TS2R     01/25/2001 18:01 01/25/2001 18:07 SU 8511/1
  ETS#OnDema01093011TS2E     01/25/2001 18:07 01/25/2001 18:07 SU 8511/1  255
  ETS#OnDema01093011TS2P     -----            -----            OH 8511/0

t1sps02:/usr/ecs/TS2/CUSTOM/utilities[63] > autorep -J ETS#OnDema01093011TS2

Job Name                        Last Start         Last End       ST   Run   Pri/Xit
____________________________ ________________ ________________ __ _______ ___

ETS#OnDema01093011TS2        01/25/2001 18:01 -----            RU 8511/1
  ETS#OnDema01093011TS2R     01/25/2001 18:01 01/25/2001 18:07 SU 8511/1
  ETS#OnDema01093011TS2E     01/25/2001 18:07 01/25/2001 18:07 SU 8511/1  255
  ETS#OnDema01093011TS2P     -----            -----            OH 8511/0

t1sps02:/usr/ecs/TS2/CUSTOM/utilities[64] >
```

Reviewing Reports (Cont.): Job Activity Report

- **Procedure**
  - Set up AutoSys
  - Type autorep -J ALL unless the command needs to be modified to:
    - specify a particular job
    - obtain a machine report
    - obtain a summary report
    - obtain a detailed report
    - obtain a query report
    - print the document
    - save the document in a file
  - Review the Job Activity Report to identify job states

Reviewing Reports (Cont.)

- **Reviewing a Job Dependency Report**
  - Production Monitor reviews a job dependency report using the AutoSys job_depends command
  - job_depends command reports information about the dependencies and conditions of jobs
    - current state of a job
    - job's dependencies
    - dependencies and nested hierarchies (for boxes) as specified in the job definition
    - forecast of what jobs will run during a given period of time

Sample Job Dependency Report

Terminal

Window Edit Options Help

```
t1sps02:/usr/ecs/TS2/CUSTOM/utilities[64] > job_depends -c -J ETS#OnDema01093011TS2E


_______________________________________________________________________________

                                                                 Start    Dependent
Job Name                         Status           Date Cond?     Cond?     Jobs?
---------                        -------       ---------------   -----    ---------
ETS#OnDema01093011TS2E           SUCCESS       No                Yes      Yes

   Condition:   success(ETS#OnDema01093011TS2R)

   Atomic Condition                              Current Status T/F
   ----------------                              -------------- ---
   SUCCESS(ETS#OnDema01093011TS2R)               SUCCESS        T

   Dependent Job Name                            Condition
   ------------------                            ---------
   ETS#OnDema01093011TS2P                        DONE(ETS#OnDema01093011TS2E)
_______________________________________________________________________________

t1sps02:/usr/ecs/TS2/CUSTOM/utilities[65] >
```

Reviewing Reports (Cont.): Job Dependency Report

- Procedure
  - Set up AutoSys
  - Type job_depends -c -J ALL unless the command needs to be modified to:
    - specify a particular job
    - obtain the current condition status
    - obtain the dependencies only
    - obtain the time dependencies
    - print the document
    - save the document in a file
  - Review the Job Dependency Repot to determine job states

Defining and Running Monitors/Browsers

- **Defining Monitors/Browsers**
  - The project does not support AutoSys monitor/browser capabilities
    - However, they are functional and available
  - Production Monitor can use the AutoSys Monitor/Browser GUI to define monitors and browsers
    - With no expectation of project support if problems are encountered
  - monitor function can limit monitoring to alarms and changes of job status (e.g., from “running” to “success” or “failure”)
  - browser function can be used to determine the eventual status of jobs run during the preceding shift or day; e.g., which jobs were successful, which jobs failed, and which jobs are still running

Sample Browser Screen

BROWSER: Browser

```
Code=0
Job: PM1DefAtt#063002000PSP STARTING   02/01/2001 16:14:29    Run# 8582:1
Job: PM1DefAtt#063002000PSP RUNNING    02/01/2001 16:14:33    Run# 8582:1
Job: PM1DefAtt#063002000PSP SUCCESS    02/01/2001 16:15:00    Run# 8582:1   Exit
Code=0
Job: PM1DefAtt#063002000PSE STARTING   02/01/2001 16:15:07    Run# 8582:1
Job: PM1DefAtt#063002000PSE RUNNING    02/01/2001 16:15:09    Run# 8582:1
Job: PM1DefAtt#063002000PSE SUCCESS    02/01/2001 16:15:12    Run# 8582:1   Exit
Code=200
Job: PM1DefAtt#063002000PSp STARTING   02/01/2001 16:15:18    Run# 8582:1
Job: PM1DefAtt#063002000PSp RUNNING    02/01/2001 16:15:21    Run# 8582:1
CHK_MAX_ALARM  Job: PM1DefAtt#063004000PSE Machine: t1spg01
  02/01/2001 16:15:24    Run# 8581:1
Job: PM1DefAtt#063002000PSE FAILURE    02/01/2001 16:15:30    Run# 8582:0   Exit
Code=0
Alarm: JOBFAILURE    Job: PM1DefAtt#063002000PSE 02/01/2001 16:15:34    Run# 858
2:0  Exit Code=0
Job: PM1DefAtt#063002000PSp SUCCESS    02/01/2001 16:15:44    Run# 8582:1   Exit
Code=1
CHK_MAX_ALARM  Job: PM1DefAtt#063002000PSE Machine: t1spg01
  02/01/2001 16:46:09    Run# 8582:1
```

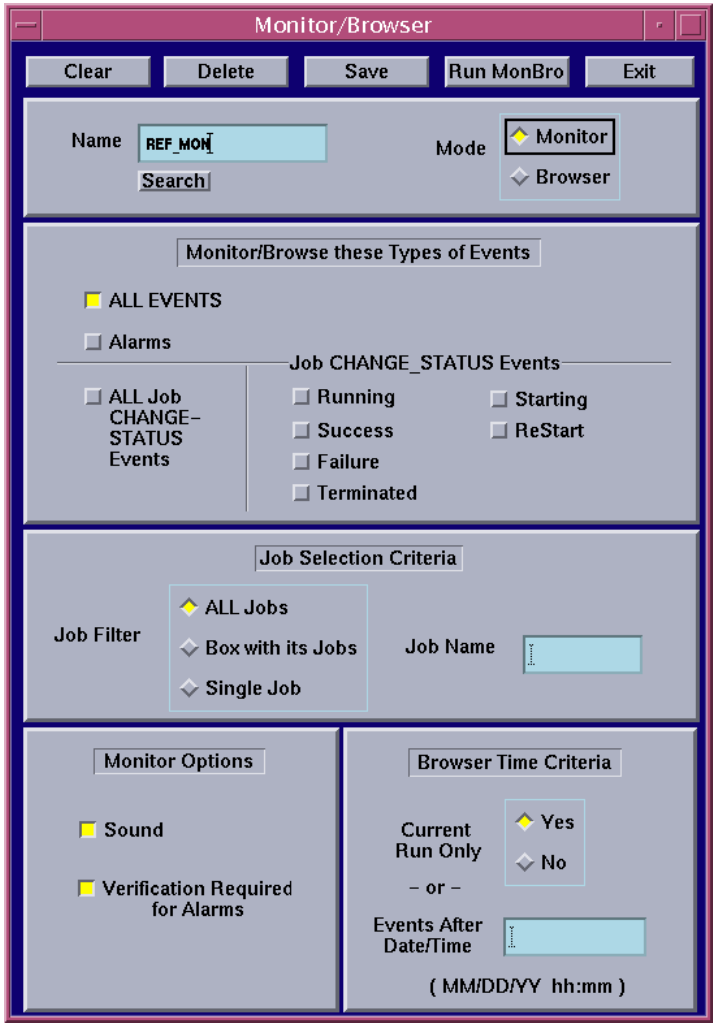
Defining and Running Monitors/Browsers

- **Defining Monitors/Browsers (Cont.)**
  - When all events for all jobs should be monitored, do not run a monitor
  - Instead, display the Event Processor log in real time (using the command autosyslog -e)
  - Running a monitor adds another connection to the database and establishes an additional process that is continually polling the database
    - That has a significant impact on system performance

Monitors/Browsers (Cont.): Defining Monitors/Browsers

- **Procedure**
  - Click on the Monitor/Browser button on the AutoSys GUI Control Panel
  - Type a name for the monitor or browser in the Name field near the top of the GUI
  - Select Types of Events
  - Select Job Status Events
  - Select the desired Job Selection Criteria
  - Select the desired Monitor Options if applicable
  - Select the desired Current Run Time and/or Events After Date/Time, which are the Browser Time Criteria if applicable
  - Select the desired Mode
  - Save the monitor/browser

Monitor/Browser GUI



Monitors/Browsers (Cont.)

- **Running Monitors/Browsers**
  - may be run from the Monitor/Browser GUI as described in the preceding procedure
  - may be run using a UNIX command

Monitors/Browsers (Cont.): Running Monitors/Browsers

- **Procedure (Running Monitors/Browsers from the Monitor/Browser GUI)**
  - Click on the Monitor/Browser button on the AutoSys GUI Control Panel
  - If the desired monitor or browser has not been previously defined, define the monitor or browser
  - Enter the name of the monitor/browser in the Name field
  - Click on the Run MonBro button
  - Click on the Exit button to exit from the Monitor/Browser GUI
  - Review the monitor/browser results
  - Type Ctrl-C in the browser/monitor window to exit from the browser or monitor

Monitors/Browsers (Cont.): Running Monitors/Browsers

- **Procedure (Running Monitors/Browsers Using UNIX Commands)**
  - **If the desired monitor or browser has not been previously defined, define the monitor or browser as described in the procedure for Defining Monitors/Browsers**
  - **Set up AutoSys**
  - **Type monbro -N *name* &**
  - **Review the monitor/browser results**
  - **Type Ctrl-C in the browser/monitor window to exit from the browser or monitor**

Tuning System Parameters

- **System parameters may be subject to control by Configuration Management (CM)**
  - **When making or requesting a change to system parameters, the CM process at the particular site must be followed (if applicable)**
- **Two databases where parameters can be set:**
  - **Configuration Registry database**
  - **PDPS database**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database**
  - **Configuration Registry Server provides a single interface (via a Sybase server) for retrieving configuration attribute-value pairs for system servers from the Configuration Registry database**
    - **When system servers are started they access the Configuration Registry database to obtain needed configuration parameters**
  - **Database Administrator has access to a Configuration Registry GUI for viewing and editing configuration data in the database**
  - **It is necessary to coordinate with the Database Administrator when changes to configuration parameters are needed**
  - **Changes to configuration-controlled parameters are subject to approval through the site CM process**

Tuning System Configuration Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.)**
  - **Default and adjusted values assigned to system parameters vary from site to site**
  - **For guidance concerning the assignment of values to parameters included in the Configuration Registry refer to document 910-TDA-022, *Custom Code Configuration Parameters for ECS***
    - **Document is available at http://cmdm ldo.raytheon.com/baseline/ under "Technical Documents"**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (parameters whose values may be modified to enhance system functioning or performance)**
  - **AppLogSize [parameter applies to all servers]**
    - **Maximum size of the application log (ALOG) file for a particular application**
    - **Recommended size varies considerably depending the nature of the application for which the file is being written**
  - **AppLogLevel [parameter applies to all servers]**
    - **Level of detail provided in the ALOG file for a particular application**
    - **Acceptable values are 0, 1, 2, or 3**
    - **A setting of "0" provides the most data**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **DebugLevel [parameter applies to all servers]**
    - **Level of detail provided in the debug log file for a particular application**
    - **Normally acceptable values are 0, 1, 2, or 3**
    - **A setting of "0" turns off logging; a setting of “3” provides a significant amount of data**
  - **DpPr_MAX_RETRIES [EcDpPrEM and EcDpPrDeletion parameter]**
    - **Number of retries (e.g., 30) to the Science Data Server for acquires/inserts before giving up**
  - **DpPr_WAIT_PERIOD [EcDpPrEM and EcDpPrDeletion parameter]**
    - **Time in seconds (e.g., 120) to wait between retries to the Science Data Server**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **DpPrRM_MAX_RETRIES [EcDpPrEM, EcDpPrGE, EcDpPrJobMgmt, EcDpPrDeletion parameter]**
    - **Maximum number (e.g., 100) of attempts to allocate a computer resource**
  - **DpPrRM_RETRY_PERIOD [EcDpPrEM, EcDpPrGE, EcDpPrJobMgmt, EcDpPrDeletion parameter]**
    - **Number of seconds (e.g., 120) between retries when trying to allocate a resource**
  - **DpPrMaxConcurrentDPRs [EcDpPrJobMgmt parameter].**
    - **Maximum allowed jobs**
    - **Three integer values (e.g., 100 100 100) are assigned to DpPrMaxConcurrentDPRs; the first for routine processing; the second for on-demand processing; and the third for reprocessing jobs**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **DpPrMinConcurrentDPRs [EcDpPrJobMgmt parameter]**
    - **Minimum allowed jobs**
    - **NOT CURRENTLY USED**
  - **DpPrAutoSysMaxDPRs [EcDpPrJobMgmt parameter]**
    - **Maximum number of completed DPRs (i.e., in SUCCESS or FAILEDPGE state) in AutoSys**
    - **When AutoSys has the maximum number of completed DPRs, the next DPR that succeeds or fails causes the oldest completed DPR to be deleted from AutoSys**
    - **If the value assigned to DpPrAutoSysMaxDPRs is too low, completed jobs are swept out of AutoSys very quickly, which may not allow the operator enough time to see that the job was completed**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **DpPrDeleteFailedPGEJobs [EcDpPrJobMgmt parameter]**
    - **If TRUE, failed PGE Jobs are removed by Job Management, as necessary, when space is needed for another job that is ready to run**
    - **If FALSE (the usual value), failed PGE Jobs are left in AutoSys**
  - **DBConnections [EcPoConnections (includes EcPlSubMgr, EcPlOdMgr, EcDpPrDeletion, EcDpPrJobMgmt and EcDpPrJobMgmtClient) parameter]**
    - **Number of connections needed by a particular application (e.g., 10 for EcPlOdMgr)**
    - **Optional parameter that specifies the number of connections to maintain in the connection pool**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **SleepDelayForFailures [EcPlSubMgr parameter]**
    - **Amount of time in seconds (e.g., 60) to wait before reprocessing failed notifications**
    - **Sleep delay used by the failed notification thread**
    - **Less frequent checking can increase speed for the other threads**
  - **SleepDelayForTimers [EcPlSubMgr parameter]**
    - **Amount of time in seconds (e.g., 60) the Subscription Manager should sleep between checking for expired timers**
    - **Should be set to the minimum amount of time a timer will be set for at the particular DAAC (min 60 sec)**
    - **Sleep delay used by the timer checking thread**
    - **Less frequent checking can increase speed for the other threads**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **SleepDelayForExp [EcPlOdMgr parameter]**
    - **Sleep delay for expiration thread in seconds (e.g., 86400)**
    - **Should be considerably greater than the sleep delay for completion threads (SleepDelayForCmp)**
  - **SleepDelayForCmp [EcPlOdMgr parameter]**
    - **Sleep delay for completion threads in seconds (e.g., 300)**
    - **Should be considerably less than the sleep delay for expiration threads (SleepDelayForExp)**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **SocketLimit [EcDpPrDeletion, EcDpPrJobMgmt, EcPlOdMgr, EcPlSubMgr parameter]**
    - **Number of connections (e.g., 200) to a server through the Hubble Space Telescope (HST) sockets middleware**
    - **Too low a number misses connections**
    - **Too high a number may adversely affect the memory of the server's host**

Modifying System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.)**
  - **When the value assigned to a parameter has been changed and saved in the Configuration Registry, the modified value does not take effect until the affected server has been restarted**
  - **Example**
    - **Debug level for the Subscription Manager log has been changed from "2" to "3" in the Configuration Registry**
    - **Modification does not affect the recording of data in the log until after a warm restart of the Subscription Manager (at which time the server would read the parameters in the Configuration Registry)**

Modifying System Parameters (Cont.)

- **Parameters in the PDPS Database**
  - The following two tables in the PDPS database have significant effects on the running of DPRs:
    - DpPrPgeLimits - controls where DPRs run
    - DpPrClassSchedulingLimits - controls how many DPRs run at a time
  - DpPrPgeLimits imposes restrictions on the number of DPRs of a particular PGE that can run simultaneously on the same virtual computer
  - A database record defines each pgeId/computerName (PGE/virtual computer) combination that will be run and how many jobs (DPRs) associated with the particular combination can run at the same time

Modifying System Parameters (Cont.)

- **Parameters in the PDPS Database (Cont.)**
  - Unless a particular host is specified when a Production Request is created, all jobs in a chain are scheduled to run on the virtual computer(s) specified for the PGE in the DpPrPgeLimits table
  - However, if no machine is specified in either the Production Request or in the DpPrPgeLimits table, the jobs run on the computer entered in the PlResourceRequirement table at PGE registration
  - An easy way to balance the load on two or more virtual computers is to specify an equal number of pgeIds to run on each virtual computer
    - If the number is large (e.g., 10,000), potentially all ready-to-run DPRs specifying the PGE can run and the number is balanced on the valid computers
    - If the number is small (e.g., two per machine), the number of DPRs using the PGE can be throttled, with the excess DPRs being queued

Modifying System Parameters (Cont.)

- **Parameters in the PDPS Database (Cont.)**
  - If controlling the total number of DPRs that can run at any one time is considered necessary, the DpPrClassSchedulingLimits table is involved
    - The table controls the total number of concurrent DPRs scheduled for Routine, Reprocessing and On-demand processing
    - When a slot is free, all ready-to-run DPRs that have empty slots in DpPrPgeLimits are considered and the DPR with the oldest time stamp in the PlDataProcessingRequest table is selected
  - The DpPrClassSchedulingLimits table has three records, one for each type of processing
    - If the DpPrClassSchedulingLimits table has no record for a particular type of processing, DPRs of that type are not allowed into AutoSys

Modifying System Parameters (Cont.)

- **Parameters in the PDPS Database (Cont.)**
  - **Values for the maxDprs and minDprs columns in the DpPrClassSchedulingLimits table are loaded at Job Management Server startup using data from the following two configuration parameters:**
    - **DpPrMaxConcurrentDPRs - maximum allowed jobs**
    - **DpPrMinConcurrentDPRs - minimum allowed jobs**
  - **Each parameter has three integer values; the first for routine processing; the second for on-demand processing; and the third for reprocessing jobs**
    - **For example, the Configuration Registry may have the following entries:**

      DpPrMaxConcurrentDPRs = 100 60 40
      DpPrMinConcurrentDPRs = 0 0 0

    - **Maximum allowed jobs is 100 for routine processing, 60 for on-demand processing, and 40 for reprocessing**
    - **Minimum allowed jobs is 0 for each type of processing**

Modifying System Parameters (Cont.)

- **Modifying the DpPrPgeLimits and DpPrClassSchedulingLimits Tables (PDPS Database)**
  - **Either the DpPrPgeLimits table or the DpPrClassSchedulingLimits table can be loaded by running the EcDpPrLoadTable.pl script from the Job Management Client tool (using the appropriate option)**
    - **The Job Management Client tool is accessed through the AutoSys Job Activity Console**
    - **The EcDpPrLoadTable.pl script loads values from an input data file**
    - **Instructions for using the script are available in the EcDpPrLoadTable.README file in the /usr/ecs/*MODE*/CUSTOM/data/DPS directory on the Queuing Server host**
    - **The same directory has a template for constructing the necessary input data file**

Modifying System Parameters (Cont.)

- **Modifying the DpPrPgeLimits and DpPrClassSchedulingLimits Tables (Cont.)**
  - The Job Management Client tool has the following options for modifying the DpPrPgeLimits table or the DpPrClassSchedulingLimits table:

    8) Change Max Concurrent Jobs for PGE Limits table

    - For DpPrPgeLimits table modifications

    9) Change Max/Min Dprs for Job Class

    - For DpPrClassSchedulingLimits table modifications.
  - For detailed instructions on modifying the tables using the Job Management Client tool refer to the procedure for Performing Job Management Client Functions
  - An alternative method of modifying the tables is to create one's own load script using SQL statements

Modifying System Parameters (Cont.)

- **Modifying the DpPrPgeLimits and DpPrClassSchedulingLimits Tables (Cont.)**
  - It is acceptable to add pgeId entries for a machine, add new machines to the DpPrPgeLimits table, or change the maximum number of DPRs that can concurrently execute in DpPrClassSchedulingLimits
  - However, values for the number of currently scheduled or running DPRs in the tables must not be changed
  - Also, note that the DpPrPgeLimits table can be empty but DpPrClassSchedulingLimits must be fully populated
    - As previously mentioned, default values for the maxDprs and minDprs columns in the DpPrClassSchedulingLimits table are loaded at Job Management Server startup using data from configuration parameters in the Registry database

Tuning System Parameters (Cont.)

- **Monitoring the Load on Processing Resources**
  - **Production Planner and Production Monitor should work with the Resource Planner to make optimum use of processing resources**
    - **Resource Planner allocates the disk partitions, CPUs, and RAM available for processing among the active modes (e.g., OPS, TS1, TS2)**
    - **Production Planner and Production Monitor monitor the load on the processing resources**

Tuning System Parameters (Cont.)

- **Monitoring the Load on Processing Resources (Cont.)**
  - **Resource Planner assigns the bulk (typically 60% - 80%) of the processing resources to the OPS mode**
    - **The remainder of the processing assets are divided among the modes used for SSI&T and new version software checkout**
  - **The Production Planner and Production Monitor monitor the load on the processing resources to identify whether the actual load is appropriately distributed among modes**
    - **They inform the Resource Planner of under- or over-use of resources as allocated**

Tuning System Parameters (Cont.)

- Monitoring the Load on Processing Resources (Cont.)
  - Disk space allocation
    - Disk space allocated to OPS mode is likely to be used to capacity
    - Disk space assigned to the other two modes may not fill up
  - CPU allocation
    - There is no one-to-one mapping of CPU allocation with actual CPUs on the science processor
    - The operating system(OS) takes care of true CPU and RAM allocation
    - Actual CPU usage during processing is limited by OS
    - If ten CPUs have been specified for a mode, only ten DPRs can be running the Execute job at a given time
    - What is really being defined is the maximum number of DPRs that will execute at a given time

Tuning System Parameters (Cont.)

- **Monitoring the Load on Processing Resources (Cont.)**
  - **CPU allocation (Cont.)**
    - CPUs can be over-allocated or under-allocated as necessary to get the most out of the CPUs on each science processor
    - If monitoring indicates that the processor is underused when OPS mode is at full processing capacity, the number of CPUs allocated to OPS mode could probably be increased
    - If the science processor is at full capacity when OPS mode is at full processing capacity (and the processor may be overworked) the number of CPUs allocated to OPS mode should be reduced
  - **Random-access memory (RAM) allocation**
    - Subject to the same considerations as CPUs
    - RAM can be over-allocated or under-allocated to get the most out of the memory on each sci. processor

Tuning System Configuration Parameters (Cont.)

- Strategies for Tuning
  - Section includes...
    - Scenario that demonstrates how DPRs might be processed under a particular set of conditions
    - Some strategies for tuning the system
  - The processing conditions include the following types of items:
    - The total number of jobs allowed into AutoSys
    - The number of CPUs available for processing
    - Characteristics of the PGEs to be processed

Tuning System Configuration Parameters (Cont.)

- **Scenario (DPR Processing)**
  - **The total number of jobs (DPRs) allowed into AutoSys is controlled by the DpPrPgeLimits table in the PDPS database**
  - **An example of some of the types of data maintained in the DpPrPgeLimits table is shown in the following table:**

computerName [Virtual Computer]	pgeId	maxConcurrent [DPRs]
A	1	20
B	1	20
A	2	20
B	2	20

Tuning System Configuration Parameters (Cont.)

- **Scenario (DPR Processing) (Cont.)**
  - **Scenario assumes that each of the virtual computers (i.e., A and B) listed in the preceding table has 16 CPUs**
    - **32 CPUs total**
  - **Relevant PGE characteristics are shown in the table that follows:**

PGE	# CPUs Used	Average Execution Time	Average Stage Time	Destage Time
1	1	5 minutes	5 minutes	5 minutes
2	1	60 minutes	5 minutes	5 minutes

  - **Scenario assumes that 100 DPRs of each type (i.e., PGE 1 and PGE 2 - 200 DPRs total) are ready to run and are released at once into AutoSys**

Tuning System Configuration Parameters (Cont.)

- **Scenario (DPR Processing) (Cont.)**
  - **Eighty (80) DPRs enter AutoSys**
  - **The remaining 120 DPRs are queued, with their assignments already made:**
    - **Machine (Virtual Computer) A**
      - **20 PGE 1s start staging**
      - **30 PGE 1s are queued on Machine A**
      - **20 PGE 2s start staging**
      - **30 PGE 2s are queued on Machine A**
    - **Machine (Virtual Computer) B**
      - **20 PGE 1s start staging**
      - **30 PGE 1s are queued on Machine B**
      - **20 PGE 2s start staging**
      - **30 PGE 2s are queued on Machine B**

Tuning System Configuration Parameters (Cont.)

- Scenario (DPR Processing) (Cont.)
  - After about five (5) minutes, all 80 DPRs that were staging have finished staging and are ready for execution
    - However, only 32 CPUs are available
  - The first 32 DPRs that ask for CPUs get them and start running
    - Sixteen (16) on Machine A and sixteen (16) on Machine B
  - Forty-eight (48) DPRs are waiting
    - Assuming that parameters in the Registry database are set as follows:
      - DpPrRM_RETRY_PERIOD = 120 seconds
      - DpPrRM_MAX_RETRIES = 100

      the waiting DPRs keep trying every two minutes for up to 100 times each before timing out (after 200 min.)
    - In this example timing out is a real possibility

Tuning System Configuration Parameters (Cont.)

- **Scenario (DPR Processing) (Cont.)**
  - **The quick jobs complete processing after five (5) minutes, freeing up sixteen (16) CPUs**
    - **In the current example, the sixteen (16) CPUs are subsequently occupied with about eight (8) five-minute PGEs and eight (8) 60-minute PGEs because CPUs are given randomly to whichever DPR gets back first to asking for them after waiting for the retry period (i.e., 120 seconds)**
    - **Priorities are not used**
    - **At first, there was a 50:50 ratio of fast:slow DPRs, now there is a 25:75 ratio of fast:slow**
    - **After another five (5) minutes, the ratio becomes 12.5:87.5 fast:slow, so 87.5 % of the CPUs are occupied by 60-minute DPRs**

Tuning System Configuration Parameters (Cont.)

- **Scenario (DPR Processing) (Cont.)**
  - **The 60-minute DPRs tend to dominate the CPUs**
    - **After one (1) hour the first batch of sixteen (16) 60-minute PGEs vacates the CPUs to be replaced by eight (8) five-minute PGEs and eight (8) 60-minute PGEs, but the five-minute PGEs become extinguished again by the slow ones**
    - **If the staging and destaging times were not the same (so the DPRs didn't have the same opportunity to hit the execution stage at the same time) the scenario would proceed differently**

Tuning System Configuration Parameters (Cont.)

- **Strategies for Tuning the System:**
  - **Limit the number of DPRs through the use of the DpPrPgeLimitsTable**
  - **Increase the declared number of CPUs for the processors to more than the actual number (overallocate CPUs)**
  - **Create new virtual computers (assigning CPUs on the processors to them) and assign (via the DpPrPgeLimits table) PGEs to run on the new virtual computers**

Tuning System Configuration Parameters (Cont.)

- Strategies for Tuning the System (Cont.):
  - Limit the number of DPRs through the use of the DpPrPgeLimitsTable
    - In the example if the number of slow DPRs allowed into AutoSys is less than the number of CPUs, there is always a channel for the fast jobs to squeeze through
    - The big disadvantage to this approach is that the slow jobs are also being prevented from staging
  - Increase the declared number of CPUs for the processors to more than the actual number (overallocate CPUs)
    - This approach allows more of each type of PGE into the science processors
    - The disadvantage is that it could overwhelm the science computers; however, they are kept busy

Tuning System Configuration Parameters (Cont.)

- **Strategies for Tuning the System (Cont.):**
  - **Create new virtual computers (assigning CPUs on the processors to them) and assign (via the DpPrPgeLimits table) PGEs to run on the new virtual computers**
    - **This approach is another way to guarantee bandwidth (CPUs) to PGEs**
    - **The disadvantage of this approach is that some CPUs could remain idle, not being seen by one of the virtual computers**
    - **In the past, there may have also been some code problems with supporting this, but those difficulties should have been resolved**

Tuning System Configuration Parameters (Cont.)

- **Strategies for Tuning (Cont.)**
  – **Probably some combination of the first two strategies is best**
    - **Increase the number of declared CPUs to be more than the total number of slow jobs allowed into AutoSys, always leaving some CPUs for a channel of fast jobs**
    - **The total number of faster-moving jobs should be increased to make sure that there is always be a queue of them available to get their channel occupied**
  – **The staging and destaging times have to be accounted for and this could change things in terms of using the DpPrPgeLimits table and the number of CPUs per processor to tune the job flow**
  – **It is important to perform regular garbage collection on all of the virtual computers**

Tuning System Configuration Parameters (Cont.)

- AutoSys Database Maintenance Time
  - Once a day the Event Processor (also known as the AutoSys daemon) goes into an internal database maintenance cycle
  - During this time, it does not process any events
  - It waits for the maintenance activities to be completed before resuming normal operations
  - Time of day for start-up of the maintenance cycle is pre-set to 3:30 AM
  - AutoSys database maintenance cycle takes approximately one minute
  - If it is necessary to change the time when the maintenance cycle occurs, whoever has “write” access to the configuration file can reset it
    - Preferably to a time when there is minimal activity

Tuning System Configuration Parameters (Cont.)

- **Procedure (Changing the AutoSys Database Maintenance Time)**
  - **Access a terminal window logged in to the Queuing Server host**
  - **Change directory to the subdirectory (e.g., autouser) containing the config.*AutoSysInstance* file**
  - **Use the vi editor to find DBMaintTime=03:30 and replace the existing time with the desired time in 24 hour format (hh:mm)**
  - **Save the edited file**

Troubleshooting Processing Problems

- **Troubleshooting:**

  process of identifying the source of problems on the basis of observed trouble symptoms

Troubleshooting Processing Problems

- Problems with production planning can usually be traced to…
  - some part of the Data Processing Subsystem
  - problems in other subsystems, including (but not necessarily limited to):
    - Planning Subsystem (PLS)
    - Data Server Subsystem (DSS)
    - Communications Subsystem (CSS)
- Fault Recovery
  - Discussed in the section on Troubleshooting Production Planning Problems

Troubleshooting Processing Problems (Cont.)

- Troubleshooting table
  - describes actions to be taken in response to some common Processing problems
  - if the problem cannot be identified and fixed without help within a reasonable period of time, call the help desk and submit a trouble ticket in accordance with site Problem Management policy

Troubleshooting Processing Problems (Cont.)

Symptom	Response
Unable to log in to the Queuing Server host (e.g., e0sps04).	Check with the Operations Controller/System Administrator to ensure that the host is "up."
GUI not displayed when the start-up script has been properly invoked.	Ensure that the DISPLAY variable was set properly. [For detailed instructions refer to the procedure for **Launching the AutoSys GUI Control Panel** (previous section of this lesson).]
Entire processing system hangs (no jobs change state over time).	1. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 2. If hosts/servers are all "up," perform the procedure for **Responding to Hanging of the Processing System** (subsequent section of this lesson).
Jobs are activated but do not get started in AutoSys.	Refer to the procedure for **Responding to Failure of Jobs to Start in AutoSys** (subsequent section of this lesson).
AutoSys box job hangs (does not change state over time).	Refer to the procedure for **Handling a Box Job that is Hanging in AutoSys** (subsequent section of this lesson).
"Preprocess" function fails (job either does not change state over time or has turned red on JobScape or TimeScape).	Refer to the procedure for **Handling a Hanging or Failed Preprocessing Job** (subsequent section of this lesson).

Troubleshooting Processing Problems (Cont.)

Symptom	Response
"Execute" job hangs (job has turned orange or oscillates between orange and green on JobScape or TimeScape).	Refer to the procedure for **Handling a Hanging Execution Job** (subsequent section of this lesson).
"Execute" job fails (job has turned red on JobScape or TimeScape).	Refer to the procedure for **Handling a Failed Execution Job** (subsequent section of this lesson).
"Postprocess" job fails (job has turned red on JobScape or TimeScape).	Refer to the procedure for **Handling a Failed Postprocessing Job** (subsequent section of this lesson).
Both the "Execute" and "Postprocess" jobs fail (jobs have both turned red on JobScape or TimeScape).	Refer to the procedure for **Handling Failure of Both Execution and Postprocessing Jobs** (subsequent section of this lesson).
On-Demand Processing Request fails.	Refer to the procedure for **Handling a Failed On-Demand Processing Request** (subsequent section of this lesson).
Other problems.	Check the log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrJobMgmt.Debug.log, EcDpPrDeletion.ALOG, *DPR*#.ALOG, *DPR*#.err, etc.) in the /usr/ecs/*MODE*/CUSTOM/logs directory for error messages. [For detailed instructions refer to the procedure for **Checking Log Files** (subsequent section of this lesson).]

Troubleshooting Processing Problems (Cont.)

- **Production Processing Troubleshooting Procedures**
  - Checking AutoSys Status
  - Checking the AutoSys Log
  - Checking Job Management Server Status
  - Checking to Determine Whether the DPR Is Waiting in the AutoSys Queue
  - Checking to Determine Whether AutoSys Is Full
  - Responding to a Condition Where a DPR Was Released But Failed Due to a JIL Failure
  - Handling Subscription Server Problems
  - Responding to a DPR That Was Released But Failed Due to an AutoSys ID Failure

Troubleshooting Processing Problems (Cont.)

- **Production Processing Troubleshooting Procedures (Cont.)**
  - Responding to a DPR That Was Released But Failed Due to Invalid DPR
  - Responding to a DPR That Was Released But Failed to Be Received by Job Management Server
  - Handling a Hanging Allocation Function
  - Running Execution Management Outside of AutoSys
  - Handling a Failed Allocation Function
  - Force-Starting a Job
  - Responding to a Restart of a Job That Fails Although All Known Problems Have Been Corrected
  - Handling a Failed Staging Function

Troubleshooting Processing Problems (Cont.)

- **Production Processing Troubleshooting Procedures (Cont.)**
  – Cleaning Up the DPS File Tables
  – Handling a Failed Preprocessing Job
  – Responding to Execution Job That Has Failed and the DPR Has Gone into "Failed-PGE" Processing
  – Handling a Failed Postprocessing Job
  – Handling Failure of Both Execution and Postprocessing Jobs
  – Handling a Failed Insertion Function
  – Handling a Failed Deallocate Function
  – Responding to a DPR that Failed in OdMgr because the PGE ID Could Not Be Found
  – Checking Log Files

Troubleshooting Processing Problems (Cont.)

- Responding to hanging of the processing system
  - Checking AutoSys status
  - Checking the AutoSys log

Troubleshooting Processing Problems (Cont.)

- **Responding to failure of jobs to start in AutoSys**
  - Checking Job Management Server status or checking to determine whether the DPR is waiting in the AutoSys queue (never got released)
    - Checking Job Management Server Status
    - Checking to Determine Whether the DPR Is Waiting in the AutoSys Queue
    - Using ISQL to Check Database Tables
    - Checking to Determine Whether AutoSys Is Full
    - Responding to a Condition Where a DPR Was Released But Failed Due to a JIL Failure
    - Handling Subscription Server Problems

Troubleshooting Processing Problems (Cont.)

- **Responding to failure of jobs to start in AutoSys (Cont.)**
  - **Responding to a DPR that was released but failed due to an AutoSys ID failure**
  - **Responding to a DPR that was released but failed due to invalid DPR**
  - **Responding to a DPR that was released but failed to be received by Job Management Server**

Troubleshooting Processing Problems (Cont.)

- Responding to a single DPS job that has failed or is hanging
  - Handling a Box Job that is Hanging in AutoSys
  - Handling a Hanging Allocation Function
  - Running Execution Management Outside of AutoSys
  - Handling a Failed Allocation Function
  - Force-Starting a Job
  - Responding to a Restart of a Job That Fails Although All Known Problems Have Been Corrected
  - Handling a Hanging Staging Function
    - Perform the Handling a Hanging Allocation Function procedure

Troubleshooting Processing Problems (Cont.)

- **Responding to a single DPS job that has failed or is hanging (Cont.)**
  - Handling a Failed Staging Function
  - Cleaning Up the DPS File Tables
  - Handling a Failed Preprocessing Job
  - Handling a Hanging Execution Job
    - Perform the Checking AutoSys Status procedure
  - Handling a Failed Execution Job
    - Perform the Checking AutoSys Status procedure
  - Responding to Execution Job and/or Postprocessing Job That Have (Has) Failed
  - Responding to Execution Job That Has Failed and the DPR Has Gone into "Failed-PGE" Processing
  - Handling a Failed Postprocessing Job

Troubleshooting Processing Problems (Cont.)

- **Responding to a single DPS job that has failed or is hanging (Cont.)**
  - **Handling Failure of Both Execution and Postprocessing Jobs**
  - **Handling a Failed Insertion Function**
  - **Handling a Failed Deallocate Function**

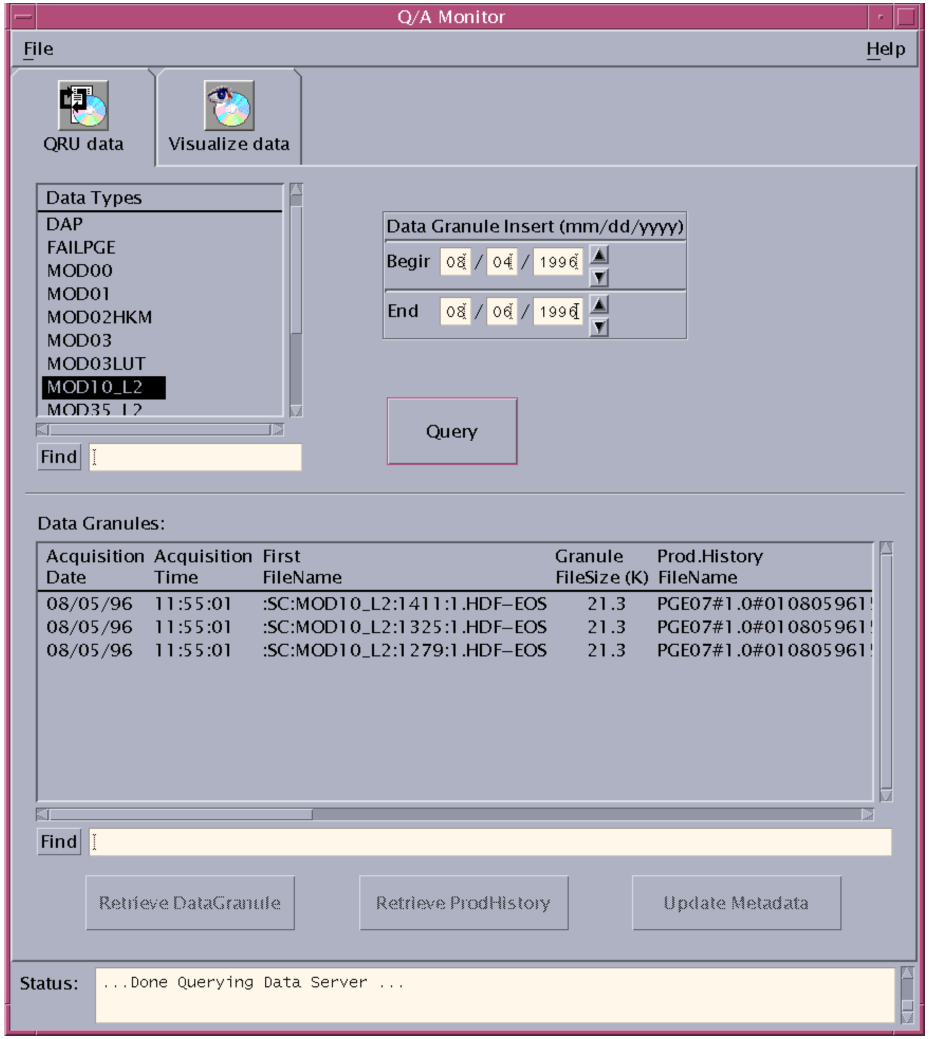
Troubleshooting Processing Problems (Cont.)

- **Handling a failed On-Demand Processing Request**
  - **Responding to a DPR that failed in OdMgr because the PGE ID could not be found**
- **Checking log files**

Launching the QA Monitor

- **Procedure**
  - **Access a terminal window logged in to the Planning/Management Workstation**
  - **Set the ECS_HOME environmental variable if necessary**
  - **Change directory to the subdirectory (e.g., utilities) containing the QA Monitor start script**
  - **Start the QA Monitor GUI in the appropriate mode**

QA Monitor GUI: QRU Data Tab



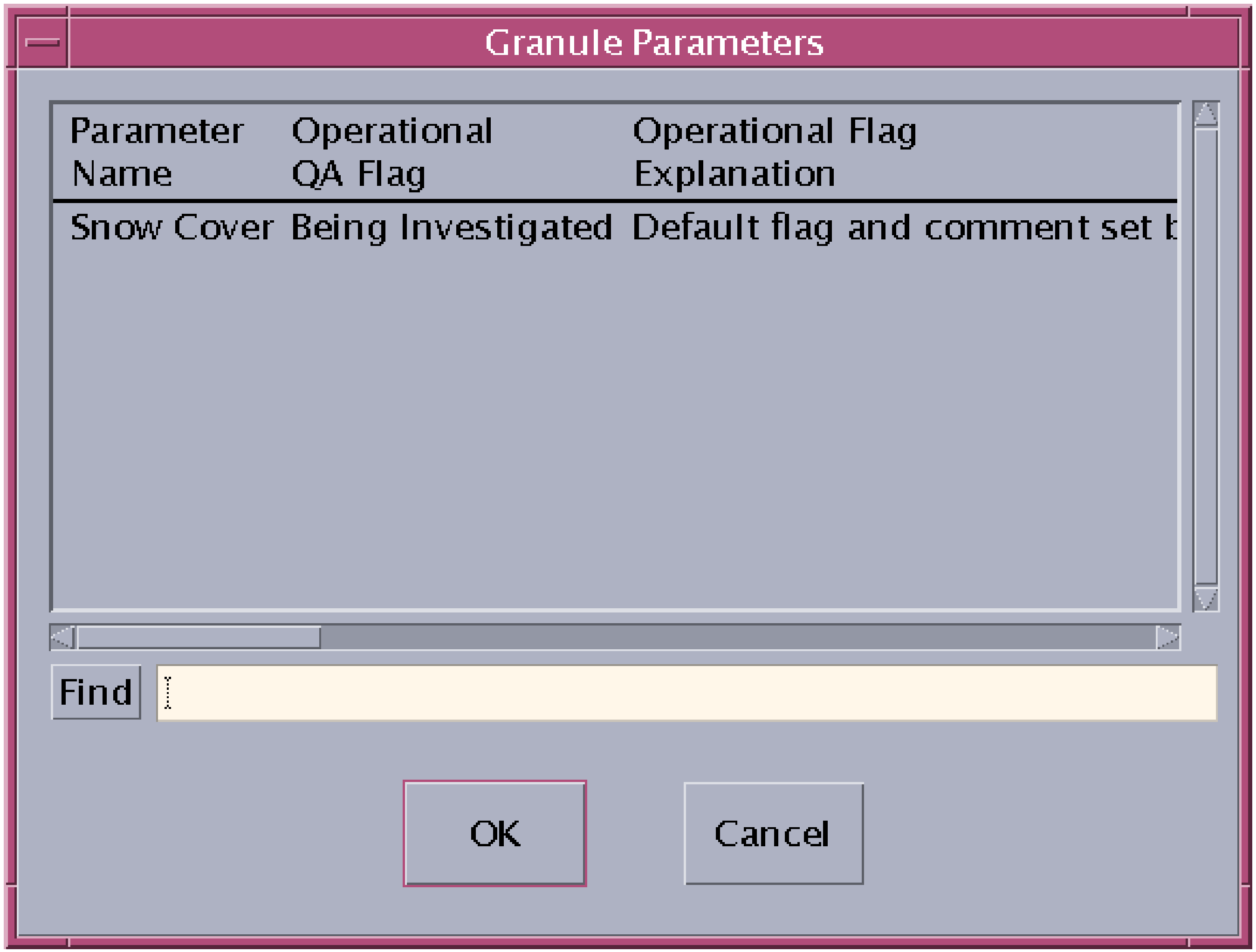
Performing Science Product Quality Assurance (QA)

- **Uses the QA Monitor application**
- **Science Computing Facility Personnel**
  - **responsible for performing QA of their products**
- **Production Monitor**
  - **updates QA metadata in response to a request from SCF personnel to set the metadata flags on specified granule(s)**

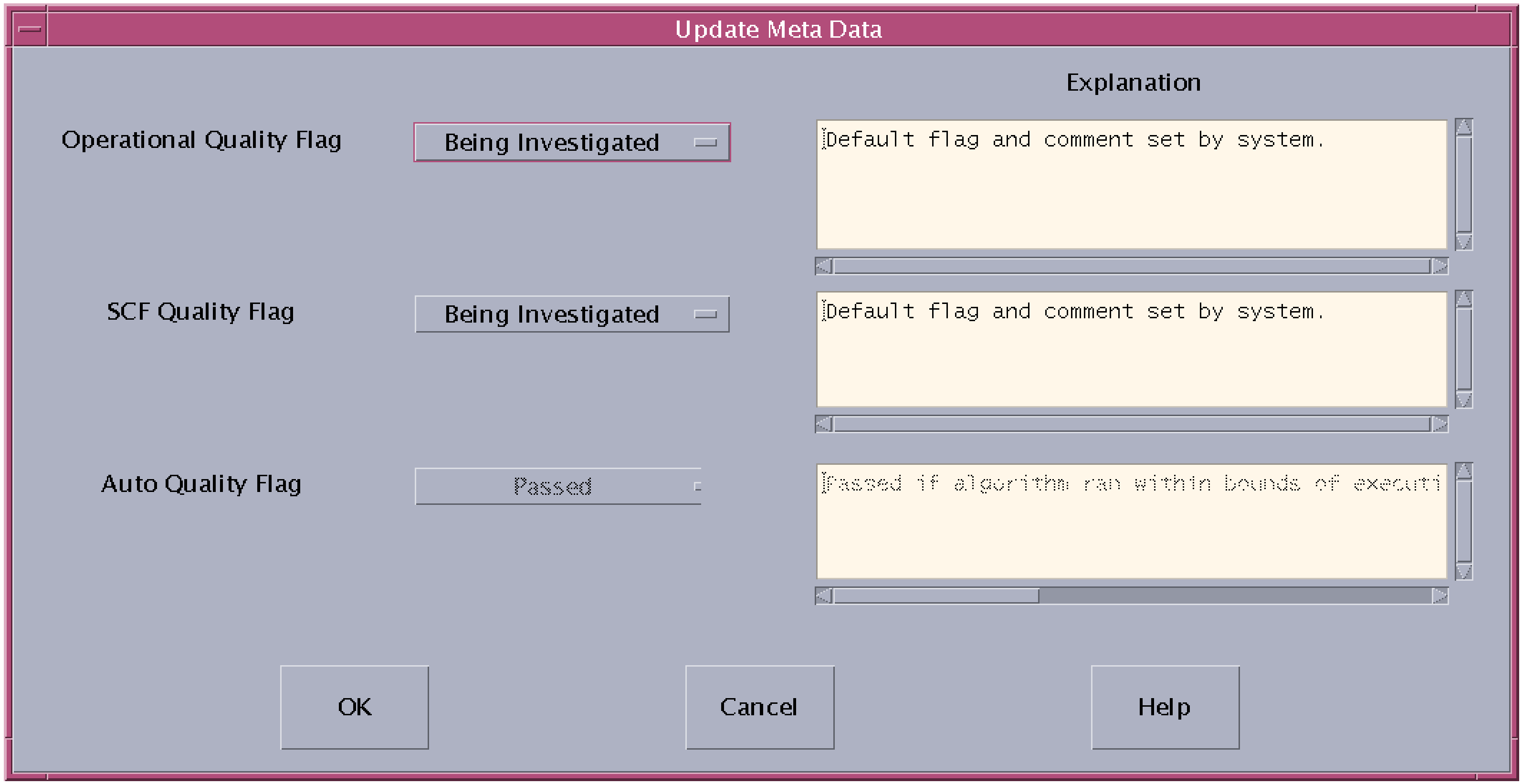
Updating Quality Assurance (QA) Metadata

- **Procedure**
  - Set up and query the database using the QA Monitor GUI
  - Select the granule with QA metadata to be updated
  - Set the operational and SCF quality flags to the appropriate value (as specified by the SCF personnel)
  - Verify that the flags have actually been set in the database by repeating the set-up and query processes

QA Monitor GUI: Granule Parameters Window



QA Monitor GUI: Update Meta Data Window



Regenerating Granules

- Produce replacements for previously generated granules that have been lost or corrupted due to failure in the archive
- General Process:
  - Retrieve the Production History file (PH) for the lost granule to determine parameters for the generation of replacement granules
  - Create Production Requests for the generation of replacement granules
  - Create and activate a Production Plan that includes the Production Requests for the generation of replacement granules
  - Prepare (if applicable) a “PDPS Residual Granules List,” which identifies granules that either cannot or should not be regenerated at the DAAC
  - Some granules do need not be reproduced; e.g., if there is a more recent version of the product

Regenerating Granules (Cont.)

- **Procedure**
  - Retrieve the Production History tar file for each granule in the Granules for PDPS Re-Generation list that needs to be reproduced
  - Launch the SSIT Manager GUI
  - Re-register the PGE (if not currently registered)
  - Launch the Production Request Editor
  - Create a Production Request for the relevant PGE/version/profile ID
  - Launch the Planning Workbench
  - Create and activate a production plan that includes the newly created Production Request(s)
  - Send the PDPS Residual Granules list to the originator of the Granules for PDPS Re-Generation list